Capacity-Maintaining Conflict Resolution for Adverse-Weather Air-Traffic Rerouting

Santosh Devasia *

* U. of Washington, Seattle, WA 98195-2600, USA (e-mail: devasia@ u.washington.edu).

Abstract: This article aims to avoid flow-capacity loss due to merges when rerouting air traffic around adverse-weather regions. However, rerouting without merges can increase the number of route intersections (potential conflicts) in the nearby sectors. To resolve such conflicts, the current work uses decentralized procedures, which can be designed in a decoupled manner without domino effects, and without the need to reduce aircraft-flow levels.

Keywords: Air Traffic Control, Conflict Resolution, Decentralized, Intelligent transportation systems

1. INTRODUCTION

This article aims to avoid flow-capacity loss that arises due to merges in current rerouting approaches used for managing adverse weather conditions, Bertsimas and Stock-Patterson [1998], Mukherjee and Hansen [2009]. Under current traffic flow management (TFM), standardized procedures in the National Severe Weather Playbook (Allen-Moorer and Weber [2004], Klein et al. [2007], Sridhar et al. [2008], Wan and Roy [2008]) allow aircraft to be rerouted around a region with adverse weather. For route simplicity, air routes (even those going to different destinations) tend to be merged before rerouting around the adverse-weather region. For example, Fig. 1a shows the ‘West Watertown’ procedure from Sridhar et al. [2008] used to merge and reroute aircraft from the west coast when a large area in the Midwest is affected by adverse weather. Merges simplify conflict resolution in nearby regions (e.g., with the route represented by a dashed line in Fig. 1) and ease the interfacing with human controllers. However, the restriction on the acceptable aircraft-flow level on the merged route leads to a reduction of the acceptable aircraft-flow levels in the routes that are merged Sridhar et al. [2008].

This article considers rerouting through nearby sectors without the use of undesirable merges, and thereby avoid-
The problem of guaranteed conflict resolution in a stable manner remains more challenging for the non-local case. For example, procedures to resolve conflicts between aircraft along intersecting routes might not be stable as shown in Mao et al. [2007]. Previous works have developed stable conflict resolution procedures for two and three intersecting routes, Mao et al. [2007]. The main difficulty is that decentralized procedures for individual intersecting routes interact with the next intersection in the sequence. Solving the resulting coupled problem, in a stable manner, can require centralized solutions, Mao et al. [2007]. In contrast, the current work demonstrates decentralized procedures that guarantee conflict resolution with multiple conflicts (intersections) by using decoupled procedures.

The current article focuses on conflict resolution along pre-specified aircraft routes as in Bayen et al. [2006], Devasia and Meyer [1999], Frazzoli et al. [2001]. Such highway-like routes, if sufficiently dense in the airspace and variable over time (Prete et al. [2008]), could provide sufficient flexibility for accommodating weather patterns (Krozel et al. [2007]), missed connections, and traffic congestion by choosing desired flight segments along the route structure in a Free-flight-like setting. Global resolution of all the intersecting conflicts in any given route structure can be achieved if the local CRPs designs are decoupled from each other. The current article demonstrates the existence of such decentralized, decoupled CRPs.

2. THE CONFLICT RESOLUTION PROCEDURE

In this section, the conflict resolution procedure is discussed.

2.1 Airspace Description

The airspace $\mathcal{AS}$, where conflicts are to be resolved (e.g., in the region surrounding the adverse weather region) is at a fixed altitude (planar flight) in which aircraft fly, with a constant speed $v_{ep}$, along one of a predefined set of distinct routes $\mathcal{R} = \{R_i\}_{i=1}^n$ with arrival points $\mathcal{A} = \{A_i\}_{i=1}^n$ and exit points $\mathcal{E} = \{E_i\}_{i=1}^n$. As illustrated in Fig. 2, each route $R_i \in \mathcal{R}$ is a directed path (not necessarily a straight line) from an arrival point $A_i \in \mathcal{A}$ to an exit point $E_i \in \mathcal{E}$.

Fig. 2. Route description. Three routes $\{R_1, R_2, R_3\}$ are shown, along with the associated arrival points $\{A_1, A_2, A_3\}$ and exit points $\{E_1, E_2, E_3\}$.

Assumption 1 [Initially Conflict Free] | Aircraft arriving at a route $R_i$ (at each arrival point $A_i$) are separated by at least distance $d$ that is greater than the minimum required separation $d_{sep}$, i.e., $d > d_{sep} > 0$.

Remark 1. Sorting of aircraft flows into a layered structure at different altitudes, with similar flight direction (such as east to west) and fixed nominal speed in each altitude, simplifies the management of air traffic. Although such layering is present in current air traffic management, conflict resolution is still needed when aircraft flows cross inside each layer—these crossings cannot be avoided due to the limited number of altitude layers available for separating flows.

2.2 CRP for Perpendicularly Intersecting Routes

Consider the conflict resolution problem for two straight-line routes $R_1, R_2$ that intersect perpendicularly as shown in Fig. 3. In general, conflict resolution can be achieved using maneuvers that change the heading, speed and altitude. However, heading changes are preferred over speed changes, which cost additional fuel for accelerating and decelerating the aircraft. Similarly, heading changes are preferred over altitude changes, which tend to incur passenger discomfort and can cause conflicts in the other altitudes, Frazzoli et al. [2001]. Therefore, this section develops a heading-change-based conflict resolution procedure (CRP) under the following simplification.

Assumption 2 [Turn Dynamics] | In the following, the turn dynamics is not modeled and heading changes are considered to be instantaneous (as in Frazzoli et al. [2001]) when designing the route modifications for conflict resolution.

Corollary 2. (Conflict-free Perpendicular Intersection). Let the spacing between aircraft in each of two perpendicularly-intersecting routes $(R_1, R_2)$ be $d$ and let the minimum separation at the intersection point be $d/2$ as shown in Fig. 3. Then, there are no conflicts between aircraft in the two routes if

$$d \geq d_{sep} = 2\sqrt{2}d_{sep}. \quad (1)$$

Proof. This follows from Iamratanakul et al. [2004], Devasia et al. [2011]

Fig. 3. Intersecting, perpendicular routes. Separation between aircraft in each route is $d$ and the minimum separation at the intersection point is $d/2$.

2.3 Conflict Resolution Procedure (CRP)

In the following, a conflict resolution procedure (CRP) for an airspace comprising of two perpendicularly intersecting routes $R_1, R_2$ (see Fig. 3) is presented when the minimal
spacings between arriving aircraft along each route is at least \( d \). Based on Corollary 2, a conflict resolution procedure can be developed for perpendicular intersections provided the separation of aircraft in each route \( d \) is greater than \( d_{sep} = \sqrt{2}d_{sep} \). However, in general, aircraft arrival spacing \( d \) might be less than \( d_{sep} \) since a spacing of \( d_{sep} < d_{sep} \) is sufficient for safety on a single route. Note that the spacing between aircraft (from a single route) can be increased by splitting the route into multiple paths. For example, splitting the route into three paths can enable the spacing on each path to be increased by three times (i.e., \( 3d \)) which is sufficiently large to develop conflict-free perpendicular intersections (from Corollary 2) since

\[
3d > 3d_{sep} > 2\sqrt{2}d_{sep} = d_{sep}.
\]

This motivates the following conflict resolution procedure for perpendicular intersecting routes – it comprises of four sub-procedures: (i) synchronize; (ii) diverge; (iii) intersect; and (iv) converge, as shown in Fig. 4 for route \( R_1 \); these sub-procedures and conditions for avoiding conflicts are discussed below.

**Synchronized Arrival** The synchronization procedure ensures that the scheduled time of arrival (STA) of aircraft at the initial way-points \( (v_1) \) for route \( R_1 \) and \( v_{22} \) for route \( R_2 \) in Fig. 4) are at discrete time instants \( t_k \)

\[
t_k = k \left( \frac{d/2}{v_{sp}} \right) = kT_{d/2} \tag{2}
\]

where \( k \) is a nonnegative even integer for route \( R_1 \) and a nonnegative odd integer for route \( R_2 \).

**Remark 3.** The time difference, \( 2T_{d/2} \) in Eq. (2), between two scheduled time of arrivals (STAs) on a single route, corresponds to the time needed to travel (with nominal speed \( v_{sp} \)) the minimum separation distance \( d \) between aircraft arriving in each route.

**Definition 1. [ETA and STA for Synchronization]**

Given an expected arrival time (ETA) \( t \) at the initial way-points of the diverge procedure \( (v_1 \) or \( v_{22} \) in Fig. 4), the synchronization procedure assigns a scheduled time of arrival (STA) \( t_k \) to the initial way-point as shown in Fig. 5. In particular, the STA \( t_k \) is chosen to be the closest (and smallest) discrete time instant to ETA \( t \), with an even integer \( k \) for route \( R_1 \) and an odd integer \( k \) for route \( R_2 \), i.e., for any integer \( k \)

\[
\text{ETA at } V_1 \in \{t_{2k+1}, t_{2k+3}\} \rightarrow \text{STA} = t_{2k+2}
\]

\[
\text{ETA at } V_{22} \in \{t_{2k}, t_{2k+2}\} \rightarrow \text{STA} = t_{2k+1}. \tag{3}
\]

![Fig. 5. Assignment of scheduled time of arrival STA at the start of the diverge procedure, for each route, based on the expected time of arrival (ETA) of each aircraft.](image)

**Remark 4.** As seen in Fig. 5, the potential STAs are separated by two discretized time points \( t_k \) defined in Eq. (2). Therefore, during synchronization, the arrival time only needs to be adjusted by a maximum of \( T_{d/2} \) (defined in Eq. 2). Thus, the distance traveled by an aircraft needs to be changed by a maximum of \( \pm d/2 \) from the nominal travel distance during synchronization.

The following synchronization procedure achieves the scheduled time of arrival (STA) by using offset maneuvers. The nominal path length (between node \( s_1 \) to node \( v_1 \) on route \( R_1 \) or between node \( v_{22} \) to node \( v_{22} \) on route \( R_2 \) in Fig. 4) is changed by \( \delta_s \), where

\[
\delta_s = (\text{STA} - \text{ETA})v_{sp}. \tag{4}
\]

by changing the distance \( x \) from way-point \( s_3 \) (in Fig. 6a) to the second heading change at \( x_1 \) is given by

\[
x = \frac{\delta_s}{2(1 - \cos(\phi_x))}. \tag{5}
\]

The offset procedure for synchronization is described for route \( R_1 \) below; the procedure is similar for route \( R_2 \) and is omitted here for brevity.

**Remark 5.** Path extension procedures are used currently in air traffic control, for example, to meter and space the aircraft arrival at airport runways.

**Path Assignment in CRP** The conflict resolution procedure (CRP) consists of splitting of each route \( (R_1, R_2) \) into three paths and choosing one of the paths for each arriving aircraft. In particular, the three paths \( \{R_{1,i}\}_{i=1}^{3} \) for route \( R_1 \) (shown in Fig. 4) are described by a set of way points \( (v_i) \):

\[
R_{1,1} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}
\]

\[
R_{1,2} = \{v_1, v_6, v_{17}, v_{18}, v_{19}, v_{20}, v_{21}, v_{15}, v_9\}
\]

\[
R_{1,3} = \{v_1, v_2, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_9\}. \tag{6}
\]
Fig. 6. (a) Synchronization procedure for route $R_1$. Distance $x$ between way-points $s_3$ and $x_1$ is given by Eq. (5). The maximum value $\Delta_s$ of $x$ corresponds to $\delta_s = d/2$ in Eq. (5). (b) Increase in path length due to $x$.

and the three paths $\{R_{2,j}\}_{j=1}^3$ for route $R_2$ are

$$R_{2,1} = \{v_{22}, v_{23}, v_{25}, v_{18}, v_{14}, v_{28}, v_{41}, v_{33}\}$$
$$R_{2,2} = \{v_{22}, v_{23}, v_{26}, v_{19}, v_{12}, v_{5}, v_{29}, v_{32}, v_{33}\}$$
$$R_{2,3} = \{v_{22}, v_{24}, v_{27}, v_{20}, v_{13}, v_{6}, v_{30}, v_{32}, v_{33}\}.$$  \hfill (7)

The path assignment procedure is illustrated in Fig. 7; the procedure is based on the index $k$ in the scheduled time of arrival (STA) $t_k$ at the initial way-points ($v_1$ or $v_{22}$).

**Definition 2. Path Allocation Procedure** Without loss of generality, it is assumed that aircraft do not arrive before time $t_0$. If STA $k$ is even (i.e., aircraft on route $R_1$ arriving at way-point $v_1$) then assign path $R_{1,j+1}$ where $j$ is $k/2$ modulus 3. If $k$ is odd (route $R_2$ at way-point $v_{22}$) then assign path $R_{2,j+1}$ where $j$ is $(k-1)/2$ modulus 3, as illustrated in Fig. 7.

![Fig. 7. Path allocation: each aircraft is assigned a path based on its scheduled time of arrival STA.](image)

**Remark 6.** The path allocation rule is cyclic and repeats after every six discrete time instants.

**Intersect Sub-Procedure** The following Lemma shows that the splitting of each route into three paths allows for a conflict-free intersection.

**Lemma 7.** (Intersection is Conflict-free). Aircraft that arrive synchronized (see Definition 1) do not have conflicts with each other in the intersection area (marked by $D_i$ in Fig. 4 for Route $R_i$) with the use of the path assignment procedure in Definition 2 if the path lengths from the arrival points to the straight line segments are all equal.

**Proof.** If the path lengths from the arrival points to the beginning of the straight line segments (e.g., from $v_1$ to $v_3$ or from $v_{22}$ to $v_{26}$ in Fig. 4) are all equal, then all possible positions of other aircraft whenever an aircraft enters a straight line segment are shown in Fig. 8. Some of the potential aircraft positions (at the discrete time instants) shown in Fig. 8 may be empty, i.e., they might not have an aircraft; however, aircraft cannot occupy any other location due to (a) arrival synchronization; and (b) equidistant path lengths to the straight line segments from the arrival points $v_1$ and $v_{22}$. Aircraft in perpendicular paths (say, $R_{1,1}$ and $R_{2,1}$ in Fig. 8) do not have conflicts since the conditions of Lemma 2 are satisfied. In particular, even when all the aircraft are present, the separation between aircraft $D$ in each path is $D = 3d > 3d_{sep}$ and the separation at the intersection point is $\bar{D}/2 = 1.5d$ (see patterns 1 and 4 in Fig. 8). There are no conflicts between aircraft on parallel paths because the paths are separated by $d > d_{sep}$.

**Diverge/Converge Sub-Procedures** The diverge and converge procedures separate and merge the routes while preserving synchronization in the different paths (for conflict avoidance at the intersection, see Lemma 7) by using equal length maneuvers as shown in Fig. 9. For example, for path $R_{1,1}$, the length from $v_1$ to $v_3$ is the same as the length for path $R_{1,3}$ from $v_1$ to $v_{10}$ via $v_2$ in Fig. 9a.

![Fig. 9. Converge (plot a) and diverge (plot b) procedures for route $R_1$ using equal-length path segments.](image)
2.4 CRPs are Decoupled and Maintain Route Flow-Capacity

Lemma 8. The CRP satisfies the following decoupling conditions:

1. **local intent** aircraft on each route \((R_1, R_2)\) exit along the same route at the corresponding exit point \(E_1, E_2\);
2. **local liveness** aircraft on each route exit the local region \(L\) within a specified bounded maximum time \(T < \infty\);
3. **local fairness** the passage through the local region \(L\) is on a first-come-first-served (FCFS) basis within each route; and
4. **local exit spacing** aircraft exiting the local region \(L\) (at each of the two exit points) are separated by at-least distance \(D\).

**Proof.** The CRP maintains the same aircraft sequence (local fairness) as at the arrival way-points \(v_1\) for route \(R_1\) and \(v_22\) for route \(R_2\) in Fig. 4) since the path lengths are the same. Since the paths for each route merge back to that route, local intent is satisfied, and the finiteness of the path length implies that the time needed to merge back to the route only takes a finite amount of time (local liveness). The discrete arrival times implies that the aircraft arrive with a minimal spacing of \(D\) — this minimal spacing is ensured at the exit as the aircraft merge back at the end of the CRP since the path lengths are the same and the nominal speed is constant.

The CRP does not change the sequence of aircraft in each route and maintains a minimal separation of \(D\) (i.e., the route-flow capacity for which the CRP is designed) at the exit. Therefore, if aircraft in one of the routes \((R_1\) or \(R_2\)) reaches another conflict point then the CRP at the second intersection point does not have to depend on the procedures used at the first CRP provided the conflict points are sufficiently separated from each other. Thus, the design of the proposed distributed CRPs (that only used local information of each route) can be decoupled from each other, without domino-type stability problems if the conflict points are sufficiently sparse in the airspace.

**Remark 9.** The local liveness and fairness conditions are not required for safety of the CRPs; however, liveness implies that aircraft will not be stuck in the airspace (e.g., in a loop) and fairness enables acceptance of the conflict resolution procedure. The first-come-first-served (FCFS) scheduling of aircraft through the airspace is considered as the canonical, fair schedule in Air Traffic Management, Erzberger [November, 1995].

**Corollary 10.** (CRP flow-capacity). Reductions in aircraft-flow levels are not needed for resolving enroute conflicts for intersecting routes with the proposed CRP.

**Proof.** This follows since each local CRP is guaranteed to maintain an exit spacing of \(d\) provided the minimal arrival spacing is \(d > d_{esp}\).

3. DISCUSSION OF IMPLEMENTATION ISSUES

Additional work is needed to study different implementation issues for the proposed CRP, such as the effect of increased time (to pass through the intersection) as well as the space required to manage such conflicts. Our current efforts aim to address these implementation issues. For example, the handling of non-perpendicular intersections and uncertainties (in aircraft speeds and arrival times) are discussed in Devasia et al. [2011].

3.1 Multiple, Close Intersections

Compound conflicts (e.g., multiple intersections in close proximity) are not considered in this article; however, the routes could be redefined so that such compound conflicts can be considered as a set of simple conflicts as studied in, e.g., Treleaven and Mao [2008] and illustrated in Fig. 10. Resolution of conflicts at each of these simple intersections can be addressed using the CRP developed in the current article.

![Fig. 10. A compound intersection of multiple routes can be rearranged into a set of simple intersections, each with two routes.](image)

When multiple conflicts occur in close proximity, the space needed for the CRP could be reduced by completing more than one intersections before the converge procedure, e.g., as illustrated in Fig. 11. Moreover, the synchronize, diverge and merge procedures (proposed in this article) could be combined with offset-type procedures for the intersections, Mao et al. [2007], Pallottino et al. [2002].

![Fig. 11. Achieving multiple intersections along route \(R_4\) before the converge procedure can reduce space needed for the CRP.](image)

3.2 Ameliorating Assumptions

Assumptions such as constant speed and instantaneous turns could be ameliorated in future work. However, the main concept to achieve decoupled decentralized CRPs, i.e., splitting of the main route into sufficient number of equal-length paths to enable route intersections without reducing the flow levels, would still be valid. For example, the effect of turn-rate limitations is studied in

The constant speed assumption is used in each local conflict resolution algorithm (in Fig. 4). It is possible, to consider different speeds for aircraft in different conflicts (with the same speed in each conflict location), which would require procedures (such as overtake protocols) to manage the flow of aircraft outside the intersection-based conflict regions. Additionally, while the current
conflict resolution algorithm is robust to small variations in the aircraft speed in each conflict resolution region (see Devasia et al. [2011]), the proposed CRP could be generalized to handle intersecting routes with different speeds. The main concepts of the current CRP would be applicable to generalized procedures such as the need (a) to split each route into multiple paths (sub-routes) with sufficient spacing to enable intersections and (b) to synchronize the arrivals.

The instantaneous turns, used in the merge and diverge procedures associated with splitting the route into multiple paths (in Fig. 4), can be generalized to include the effects of aircraft turn dynamics. This would necessitate, for example, the consideration of continuous turns instead of instantaneous turns. Turns in the algorithm (during merge, diverge and synchronize) would have to be modified to include the aircraft dynamics. Our ongoing work is aimed at quantifying the arrival spacing conditions as well as the potential increase in the local space needed for the CRP with the inclusion of aircraft turn dynamics. In particular, the effect of finite turn-rates is studied in Yoo and Devasia [2011], which also evaluates the applicability of the proposed approach for conflict resolution in the Cleveland sector of the US airspace.

ACKNOWLEDGEMENTS

Fruitful discussions with George Meyer and Gano Chatterji from the NASA Ames Research Center and Dhanakorn Iamratanakul from the U. of Washington are gratefully acknowledged.

REFERENCES


