AUTOMATED CONFLICT RESOLUTION, ARRIVAL MANAGEMENT AND WEATHER AVOIDANCE FOR ATM

H. Erzberger, Todd A. Lauderdale, Yung-Cheng Chu

***NASA Ames Research Center**
On Assignement from
University of California, Santa Cruz
Heinz.Erzberger@nasa.gov

****NASA Ames Research Center**
Todd.A.Lauderdale@nasa.gov

*****Aerospace Computing Inc.**
Yung-Cheng.Chu-1@nasa.gov

Keywords: *Separation Assurance, Air Traffic Management*

Abstract

The paper describes a unified solution to three types of separation assurance problems that occur in en-route airspace: separation conflicts, arrival sequencing, and weather-cell avoidance. Algorithms for solving these problems play a key role in the design of future air traffic management systems such as NextGen. Because these problems can arise simultaneously in any combination, it is necessary to develop integrated algorithms for solving them. A unified and comprehensive solution to these problems provides the foundation for a future air traffic management system that requires a high level of automation in separation assurance. The paper describes the three algorithms developed for solving each problem and then shows how they are used sequentially to solve any combination of these problems. The first algorithm resolves loss-of-separation conflicts and is an evolution of an algorithm described in an earlier paper. The new version generates multiple resolutions for each conflict and then selects the one giving the least delay. Two new algorithms, one for sequencing and merging of arrival traffic, referred to as the Arrival Manager, and the other for weather-cell avoidance are the major focus of the paper. Because these three problems constitute a substantial fraction of the workload of en-route controllers, integrated algorithms to solve them is a basic requirement for automated separation assurance. The paper also reviews the Advanced Airspace Concept, a proposed design for a ground-based system that postulates redundant systems for separation assurance in order to achieve both high levels of safety and airspace capacity. It is proposed that automated separation assurance be introduced operationally in several steps, each step reducing controller workload further while increasing airspace capacity. A fast time simulation was used to determine performance statistics of the algorithm at up to 3 times current traffic levels.

1 Introduction

The design of the future U.S. air traffic system, referred to as NextGen, is the subject of current research at universities and research centers around the country. Although system concepts continue to evolve as research progresses, it is likely that the design will include higher levels

of automation in separation assurance. If such an approach were adopted, it would be a significant change from the current method of separation assurance, which is largely performed by controllers. The major challenge in designing a system that depends on automation for making safety-critical decisions is demonstrating that it can be made as safe or safer than the current method over a wide range of operating conditions and traffic densities. A postulated system, referred to as the Advanced Airspace Concept (AAC), addresses the safety issues associated with automated separation assurance by incorporating two independent systems for conflict detection and resolution, each of which is designed to act over separate time ranges [1]. One element of this system is designed to handle conflicts predicted to occur in the range of approximately 2 to 20 minutes. It would be the mainstay for solving separation assurance and related problems including arrival sequencing and weather-cell avoidance. This element can be considered the strategic problem solver in AAC and is referred to as the Autoresolver. Its design and performance for resolving conflicts in en-route airspace are described in several papers [2, 3]. The new functions as well as performance enhancements of existing functions described in this paper have been integrated into the Autoresolver suite of software.

The second separation-assurance element in AAC focuses exclusively on handling tactical conflicts, defined as those with times to loss of separation of less than two minutes. Its main purpose is to provide a safety net for those infrequently occurring situations when conflicts are not detected and/or resolved strategically in a timely manner by the Autoresolver. This element is referred to by the acronym TSAFE, which stands for Tactical Separation Assured Flight Environment. It should not be confused with the Traffic Alert and Collision Avoidance System, TCAS, which is an in-service system for collision avoidance installed in commercial airliners. The algorithmic design and operational concept for TSAFE is given in [4]. Although the algorithms incorporated in these two elements perform similar functions, they differ substantially in their analytical formulation as well as in their software and operational implementation.

Operational implementation of a system for automated separation assurance requires an air-ground data link that allows the ground-based AAC systems to uplink solution trajectories directly into systems onboard aircraft. Furthermore, in order to achieve the full safety benefits of the two independent separation-assurance elements in AAC, it is also desirable to have available two independent data links for exclusive use by each of the two elements, one dedicated to up linking strategic resolutions and a separate one dedicated to uplinking tactical or TSAFE resolutions. Such independent data links will become available in the next few years. A conventional voice link will continue to be required in order for controllers to accommodate unequipped aircraft and to handle emergency situations for all types of aircraft.

A simplified diagram of the elements comprising the architecture of the AAC is shown in Fig. 1. It should be noted that in the diagram TCAS is also assumed to be a part of the complete separation- assurance system. A fault-tree analysis carried out for this system architecture suggests that it would have an acceptably low probability of collision risk at up to three times current levels [5].

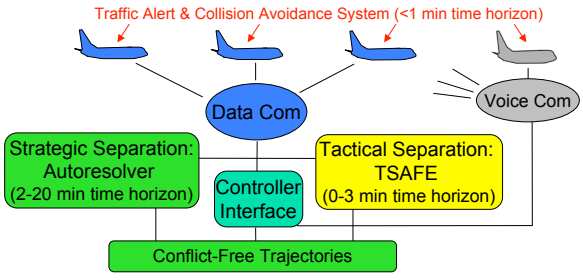


Fig. 1 Architecture of Advanced Airspace Concept.

2 Review of Existing Autoresolver Design and Feedback from Users

Initially, the Autoresolver was developed to be a software system for resolving loss-of-separation

conflicts for all types of traffic, including arrivals. The design of the algorithm and initial performance results are described in [2]. Farley [3] and Kupfer [6] conducted extensive performance evaluations of the algorithm for a range of traffic loads and several en-route centers using fast-time simulations. Prevot [7] used the Autoresolver software to conduct human-in-the-loop simulations that focused on evaluating automated separation-assurance concepts for future air- traffic-management systems.

Before describing the new functions incorporated into the Autoresolver, it is instructive to briefly review the original design approach, which remains largely unchanged. The input to the Autoresolver is a list of conflicts generated by a separate system, referred to as a Conflict Probe. An analytical formulation that forms the basis for a Conflict Probe algorithm and its software implementation is given in [8]. A Conflict Probe detects future loss of separation, defined as two aircraft predicted to come within 5 nmi horizontally and 1000 feet vertically. The Conflict Probe updates the conflict list at a cyclic rate synchronized to the update cycle of the surveillance system, typically every 12 seconds in en-route airspace. The list of aircraft conflict pairs is usually ordered by predicted time to first loss of separation, where earliest predicted conflicts are at the top of the list. The Autoresolver then attempts to resolve new conflicts on the list in the priority of time to first loss. Resolutions of conflicts with times to first loss greater than a threshold time are deferred to a later update cycle. Threshold times range from 8 minutes for purely en-route conflicts to 20 minutes for arrivals. These are nominal times that depend on the prediction accuracy of the Conflict Probe as well as operational experience.

In addition to times to first loss, the Autoresolver uses detailed characteristics of the conflict encounter in its resolution logic. These characteristics, which are also provided by the Conflict Probe, include aircraft types, current aircraft location coordinates, flight plans, speeds, encounter angles, and airspace boundaries, as well as other information that must be considered in generating a resolution trajectory. First, procedural logic, analogous to an expert system, determines a prioritized set of trial-resolution maneuvers, which, on the basis of controller experience, are most likely to result in a successful resolution trajectory. Second, the Autoresolver uses analytical formulas and heuristics to calculate the parameters of a simplified resolution trajectory for each of the maneuvers in the set. The simplified trajectory includes a segment that returns the maneuvered aircraft back to its original flight plan at a reasonable distance downrange of the conflict. The parameters that specify the simplified trajectory are chosen to provide all the information that a trajectory engine needs to determine a resolution trajectory that can be flown accurately by a given aircraft type.

In the third step, the Autoresolver sends the simplified trajectory parameters to a trajectory engine that computes a trial resolution trajectory. A trajectory engine is a complex and computationally intensive software system that simulates in fast time the flight of an aircraft using detailed models of aircraft dynamics and performance. The Autoresolver has been designed to be independent of a specific trajectory engine and has worked successfully with three different engines in use at NASA Ames Research Center.

In the fourth step, the Autoresolver sends the trial resolution trajectory to a Conflict Probe that determines if the trajectory solves the conflict and does not introduce new conflicts unintentionally. Such unintentional conflicts are referred to as secondary trajectory conflicts and, if found, are cause for rejecting the trajectory. To account for trajectory prediction uncertainty an additional horizontal buffer, nominally 2 nautical miles, is added during this conflict detection process.

Thus, at this stage of the resolution process a critical decision point has been reached. If the trial trajectory resolves the primary conflict and has not introduced secondary conflicts, the Autoresolver has successfully generated a candidate resolution trajectory. This trajectory can be issued to the conflict aircraft by a controller or automated agent. If, however, the trajectory has failed to resolve the conflict, the Autoresolver re-

turns to the second step and starts the resolution process again using the next trial maneuver in the set. This iterative process continues until a successful resolution trajectory is found or until all available options have been exhausted. In the unlikely latter case, certain fallback options come into play as described in [2].

For en-route conflicts with times to first loss between 2 and 8 minutes, the Autoresolver requires only a few trial iterations to find a successful resolution trajectory. Simulation tests show a gradual increase in the number of iterations with increasing traffic while still resolving all detected conflicts at levels up to 3 times current traffic levels. Further details on the design of the original Autoresolver can be found in [2].

Feedback from experimental users and evaluators of the Autoresolver revealed both strengths and weaknesses of the initial version of the algorithm. While users were generally satisfied with the ability of the Autoresolver to generate resolution trajectories for all types of conflicts at up to 3 times current traffic levels, they requested that the Autoresolver be revised to routinely supply multiple types of resolutions for each conflict when possible. Experimenters conducting human-in-the-loop simulations using the Autoresolver discovered that controllers and pilots wanted the flexibility to choose from horizontal, vertical, and speed resolutions. Users also pointed out the need for trajectories that avoid convective weather cells. The combination of high traffic density and convective weather can create complex re-routing problems. Automation should be designed to solve these problems without creating conflicts. Finally, a new algorithm that can efficiently manage arrival traffic subject to time-based metering and in-trail spacing constraints was needed.

3 New Autoresolver

These additional requirements led to the development of the new Autoresolver that comprises the two new algorithms and modifications to the original algorithm. Together they comprise the upgraded Autoresolver described in this paper. A diagram of the new Autoresolver with blocks representing the algorithms that generate solutions for the three problems is shown in Fig. 2. Also shown in the figure are the primary inputs to the algorithms from external sources. All three algorithms interact with the Trial Trajectory Engine and Trial Conflict Probe in the process of generating resolution trajectories.

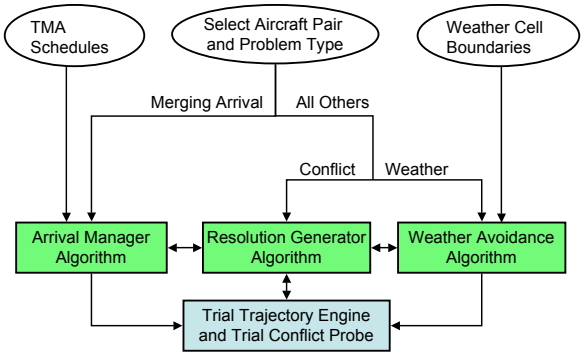


Fig. 2 Functional Diagram of Autoresolver.

4 Resolution Generator

Instead of stopping the resolution process after finding the first successful resolution as originally designed, the resolution algorithm was modified to continue searching for additional resolutions using eligible alternatives in horizontal, altitude, and speed maneuvers. Furthermore, the search for additional resolutions was extended to both aircraft in the conflict pair. Clearly, there was a risk that the additional search for resolutions could result in unacceptable increases in computational workload and could therefore exceed the maximum time allowed to complete the resolution process. However, evaluations of the modified algorithm in both fast and real-time simulations have shown that the time to perform the computations remains within acceptable limits.

The generation of multiple resolutions for each conflict has made it possible to optimize the choice of resolution trajectories. An economically important performance criterion to be minimized is the time delay introduced by a resolution trajectory. Thus, the algorithm compares all

available trajectories and chooses the one with the smallest time delay. As in the original version, the algorithm retains a set of rules for excluding specific maneuver types and/or specific aircraft from being maneuvered under defined conditions. However, the algorithm gains the flexibility to use non-preferred resolution trajectories in circumstances where these trajectories are the only ones available to resolve the conflict. The multiple resolutions, illustrated in Fig. 3, comprise several types of horizontal path, altitude, and speed resolutions. Horizontal trajectories yield the greatest number of new resolution types and are illustrated in Fig. 3(a).

The first type of horizontal resolution is called Direct-To. In previous work Direct-To was developed as a decision support tool for controllers to help resolve conflicts and improve flight efficiency [9]. It attempts to solve a conflict by taking an aircraft directly to a downstream waypoint, thus bypassing a dogleg in the flight plan. This type can only be successful if there is a dogleg to bypass. Because doglegs are relatively common in the current route structure, experiments have shown that Direct-To resolutions succeed at a surprisingly high rate. The probability of success is further increased when both aircraft are eligible for Direct-To solutions. On rare occasions a Direct-To resolution has been found for both conflict aircraft. A Direct-To resolution is highly desirable since it yields a net time saving (negative time delay) for the conflict aircraft that is assigned to perform it. It is usually the winner of the minimum-delay comparison test for the best resolution.

A path-stretch method, referred to as the elliptic path-stretch algorithm, provides the second type of horizontal resolution trajectories. This method has proven to yield the greatest success rate of all horizontal resolution methods. While effective for all types of conflict encounters, it is by far the most important maneuver used for handling arrival aircraft. The algorithm attempts to generate path-stretch resolutions for each conflict aircraft in both left- and right-turn directions relative to the current aircraft heading. This process can yield up to four resolution trajectories. Ana-

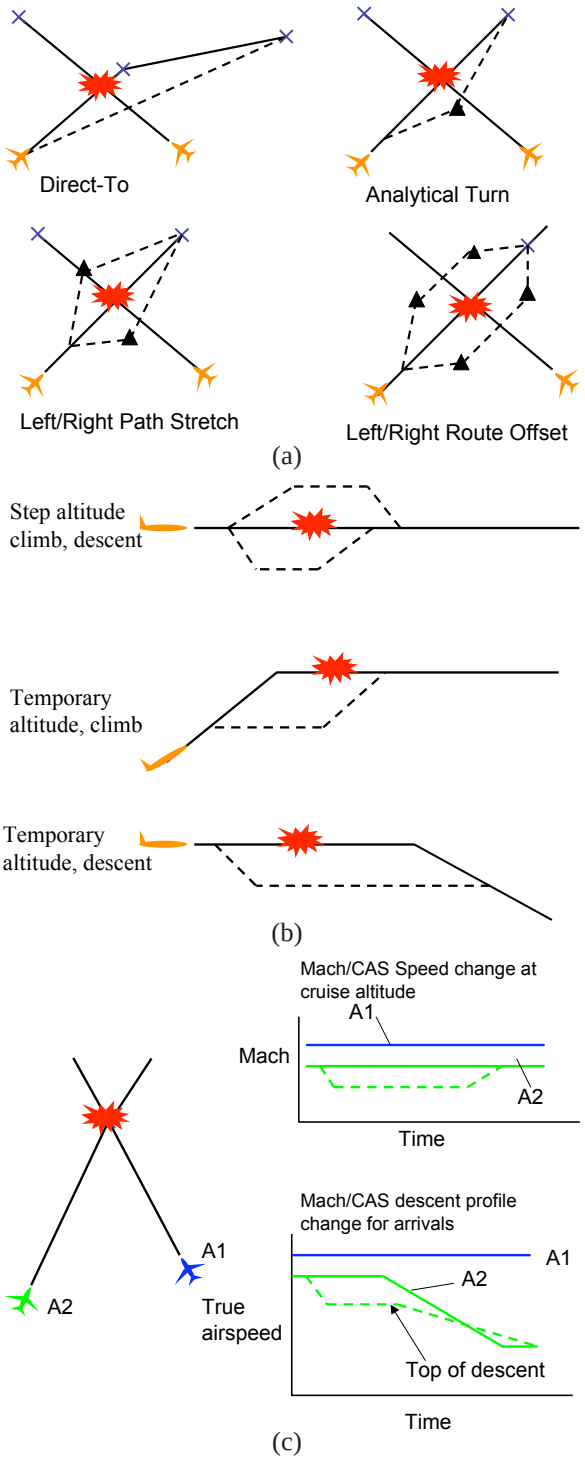


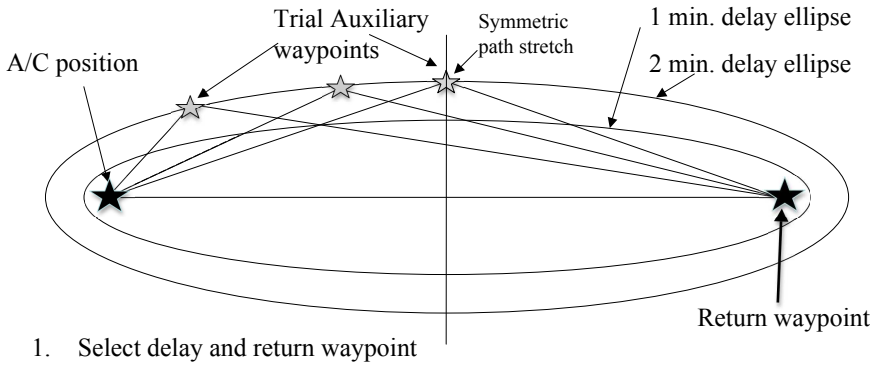
Fig. 3 Resolution trajectories: (a) Horiztonal resolution types; (b) Altitude resolution types; (c) Speed resolution types.

lytical formulas for the algorithm can be found in [2].

Elliptic path stretches play a critically important role in the performance of the Autoresolver and are therefore briefly reviewed here with the help of Fig. 4. They are characterized by two parameters; the first is a specified delay and the second is the vector angle of the maneuver. These two parameters define the location of the auxiliary waypoint as shown in Fig. 4. The word elliptic refers to the fact that the locus of waypoints for a specified value of delay traces out an ellipse whose two foci are the initial location of the aircraft and the location of the return waypoint. This formulation has the further advantage of yielding an analytical formula for the coordinates of the auxiliary waypoint as a function of the two parameters. The formula thus yields all possible path stretches that generate a specified amount of delay. It is used for generating sets of arrival trajectories that are required to cross an arrival fix at a specified time or with a specified delay. The second degree of freedom provided by the vector-angle parameter in the formula is primarily used to help resolve secondary conflicts while holding the delay constant. Note, however, that the path stretches calculated by the algorithm do not explicitly resolve a conflict. The Trial Trajectory Engine and Trial Conflict Probe are required to first generate and then check the trajectory corresponding to a set of path-stretch parameters for conflicts. Several iterations on these parameters are generally required to clear a conflict. The iteration starts with the specified delay and a vector angle yielding a symmetric path stretch (see Fig. 4). Then, if secondary conflicts are encountered, the vector angle is steeped in increments. Thus, the elliptic path-stretch formulation used in combination with the Trial Trajectory Engine and Trial Conflict Probe provides a method for computing trajectories that are simultaneously conflict-free and meet a specified arrival time. For en-route conflicts that are not constrained by arrival time, the two degrees of freedom in the algorithm help find a resolution with the least delay that resolves a primary conflict while also avoiding secondary conflicts.

The analytical-turn algorithm calculates a simplified resolution trajectory that is designed to achieve a specified minimum separation distance between the two primary conflict aircraft [2, 10]. It uses an analytical solution that determines the heading change required to achieve a specified minimum separation distance. Aircraft turn rate is not modeled in the derivation. Furthermore, assumptions made in the derivation of the solution limit its application to encounters where both conflict aircraft are flying at constant altitude, speed and heading toward the conflict point. Thus, trajectories that contain doglegs and altitude changes near the area of the conflict can cause the algorithm to fail. Together, these properties are referred to as eligibility conditions. To avoid its misapplication, the algorithm first checks the current trajectories of the conflict aircraft to determine if they meet the constant speed, altitude and heading conditions. If they do, the algorithm calculates all possible maneuvers that achieve the specified minimum separation distance; if not, the algorithm is bypassed in favor of the elliptic path stretch method. The solution giving the least delay is then further evaluated by the Trial Trajectory Generator and Trial Conflict Probe to ensure that the primary conflict is resolved and that no secondary conflicts have been introduced. Thus, a successful resolution obtained from this algorithm is time-efficient and likely to win the contest for the minimum-time-delay resolution, unless a Direct-To resolution is also available. However, simulations have shown that the algorithm may introduce unsolved secondary conflicts or is bypassed because the conflict aircraft fail to meet the eligibility conditions described earlier. As a result, the analytical turn algorithm contributes only a small percentage to the winners among horizontal resolutions types in the regions tested. The utilization rate of the algorithm is likely to be greater in airspace where flight plans of aircraft contain long straight segments.

Route offset is the most recent addition to the set of horizontal-resolution maneuvers implemented in the algorithm. It consists of a segment that is parallel to the current route of flight of a



1. Select delay and return waypoint
2. Iterate on trial auxiliary waypoint location to selected delay ellipse until conflict free solution is found
3. If no solutions are found, repeat step 2 for higher delay ellipse

Fig. 4 Constant delay elliptic path stretch algorithm.

conflict aircraft and displaced from it by a specified distance. The offset distance is typically chosen as 10 nmi. The conflict aircraft captures the offset segment via a transition vector nominally chosen as 30 degrees relative the current heading of the aircraft. The resulting intersection point defines the coordinates of the first waypoint. The length of the offset segment is increased in 50-nmi increments up to a maximum of 200 nmi until one is found that resolves the conflict. The Trial Trajectory Engine and Trial Conflict Probe must evaluate each trial offset maneuver to determine if the conflict has been cleared. A waypoint located at the end of the offset segment specifies the start of a recapture maneuver that takes the aircraft back to its original route. A conveniently located downstream waypoint on the existing route is chosen as the return waypoint that terminates the transition segment. The offset maneuver is limited to conflict aircraft that are on a straight-line route segment for at least 100 nmi. Both left- and right-sided offset maneuvers are attempted for each eligible aircraft. Users of the algorithm can change the above default parameters values to fit the application.

Navigation systems with built-in route offset capabilities, available on some aircraft, are the primary motivation for using these maneuvers in conflict resolution. A controller can issue an offset maneuver to a pilot by specifying only the offset distance and its length. The pilot then easily enters these parameters into the navigation system. The convenience of this procedure is important in an operational environment where a data link is not available to communicate strings of numerical data such as the latitude and longitude of auxiliary waypoints. In simulation the offset maneuvers have proven to be effective in resolving certain types of eligible conflicts. For example, they can be effective for resolving in-trail overtake conflicts, where they provide a passing lane for faster aircraft to pass a slower aircraft. The offset maneuver could be given preference over other options if the controller has to issue the resolution via voice communications rather than data link.

Avoiding penetration of convective weather cells is another constraint included in determining acceptable horizontal resolutions. Generally, the Autoresolver resolves a weather conflict prior to loss of separation conflicts. A separate algorithm (described in a later section) is designed specifically to avoid weather cells when they are the primary problem to be resolved. However, certain types of weather conflicts must be handled during the process of resolving loss-of-separation conflicts. Thus, it is possible that a trial horizontal resolution that resolves the primary conflict induces a new conflict with a weather cell. In that case, the al-

gorithm rejects such a resolution and proceeds to generate additional trial resolutions in an attempt to eliminate both types. This process is similar to the way the algorithm handles induced secondary loss-of-separation conflicts. Another special case involves a weather-cell conflict that is detected down range of a close-in primary loss-of-separation conflict. Thus, for conflicts within 4 minutes to loss-of-separation followed by a down-stream weather conflict, the Resolution Generator gives priority to resolving the loss-of-separation conflict, ignoring the weather conflict. Occasionally, the weather conflict is resolved incidentally to the maneuver resolving the loss-of-separation conflict. That is referred to as a bonus resolution. If, however, the weather conflict remains unresolved, it is sent to the Weather Avoidance Algorithm for resolution. The Weather Avoidance Algorithm starts at the point on the trajectory that is clear of the loss-of-separation conflict. The final trajectory will be a concatenation of two maneuvers, called a compound maneuver.

Fig. 3(b) illustrates the three types of altitude resolutions the algorithm attempts to generate for each conflict. These types were included in the original version of the algorithm and are described in detail in [2]. Note that the step altitude resolutions return the aircraft to its original flight-plan altitude after aircraft are clear of the conflict zone. The algorithm chooses an altitude resolution type that is compatible with the flight phase of an aircraft. Thus, it assigns the first type to an aircraft in cruise, the second to an aircraft during climb, and the third to an arrival aircraft that is close to its top of descent. Since the flight phases of the two conflict aircraft generally differ from each other, successful altitude resolutions for the two conflict aircraft can be any two combinations of these three types. For example, one aircraft in cruise and the other in a climb can yield up to three successful altitude resolutions. Pilot subjects in simulations pointed out the need for alternative altitude resolutions. When presented with a single type of altitude resolution such as a step climb generated by the original version of the algorithm, pilots often inquired if a step descent was available. There are various reasons for such requests, including a desire by a pilot to avoid turbulence and/or optimize fuel efficiency, factors not considered in the algorithm decision process.

Speed-change resolutions complete the repertoire of resolution types the algorithm attempts to generate for each conflict. The algorithm can generate two types of speed resolution, as illustrated in Fig. 3(c). In one type the aircraft whose speed is changed returns to the original speed after the conflict has been cleared. In the second type, applicable only for arrival aircraft that are close to the top of descent, the aircraft starts the speed change in cruise and maintains the change for the duration of the descent. Descent speed profiles consist of a constant Mach segment starting at the top of descent followed by a constant calibrated airspeed (CAS) segment that continues to the bottom of the descent. The algorithm iterates on the Mach/CAS parameters in combination with the Trial Trajectory Engine and Trial Conflict Probe to search for a successful resolution. For slow descents only a constant CAS segment may be necessary. These two types are unchanged from the original version of the algorithm. Similar to the handling of the altitude resolution types, the algorithm assigns a type based on the flight phase of the aircraft. Simulation tests have shown that attempted speed resolutions for aircraft in cruise flight seldom succeed and, even if they do, they are usually not the recommended resolution. However, the second type is effective for resolving conflicts between arrivals and is frequently the first choice among successful resolutions.

Two additional resolution types (not shown in Fig. 3) are occasionally used in specific situations to solve arrival problems. First, a compound maneuver consisting of a path stretch and a descent speed reduction is used when a large delay is required to fit an arrival into a specified time slot. Second, for an aircraft currently in climb that has departed from a nearby airport and must be merged into an arrival stream, a reduction in cruise altitude is used for delay and conflict resolution. The reduction in average true airspeed

resulting from the reduction in cruise altitude introduces delay into the trajectory that can help to resolve merging conflicts with other arrivals near the top-of-descent point.

The final step of the algorithm is to establish a preference order over the set of successful resolutions. While the primary criterion for the top choice is the resolution with the least delay, this criterion can be overruled by other factors, the most important of which are rules for choosing the preferred maneuver aircraft. Thus, in conflicts between arrivals and nonarrivals, nonarrivals are generally preferred over arrivals. An exception to this rule is made for an arrival that is close to its top of descent and for which a descent to a temporary altitude is in the set of successful resolutions. Another rule gives preference to an aircraft not previously maneuvered for resolving a conflict over an aircraft that recently performed a resolution maneuver. Preference is also given to maneuvering a climbing aircraft in conflict with an overflight. Finally, an aircraft close to an airspace boundary is usually exempted from performing a maneuver. Additional maneuver preference rules applicable to handling arrival conflicts are discussed in the section on the Arrival Manager. Thus, the algorithm chooses the minimum time resolution that does not violate the previously discussed preference rules. However, if no resolutions that obey these rules are available, then, as a last resort, the Autoresolver chooses the best available resolution that maneuvers a nonpreferred aircraft. Similarly, the algorithm chooses an operationally excluded maneuver type for a nonpreferred aircraft if that is the only resolution available that will resolve the conflict.

Holding maneuvers are sometimes used to solve certain types of traffic management problems. While holding is a general tool for traffic management, it is not considered a resolution maneuver type and is therefore not included among the available resolution maneuvers. Holding could be added to handle special situations in future versions of the Autoresolver.

Fig. 5 lists the maximum number of trial trajectories the Resolution Generator can attempt for each type of resolution, the maximum number of successful resolutions for each type and the categories of flights eligible for each type. The search for each type of resolution starts with the most desirable resolution of that type and often succeeds on the first trial. Additional trials are usually needed only if secondary conflicts are encountered. Path-stretch resolutions contribute the most trial trajectories, since they require a search over two sets of discretized parameters. Only in complex problems involving multiple secondary conflicts are a large number of trials required to find a successful path stretch resolution.

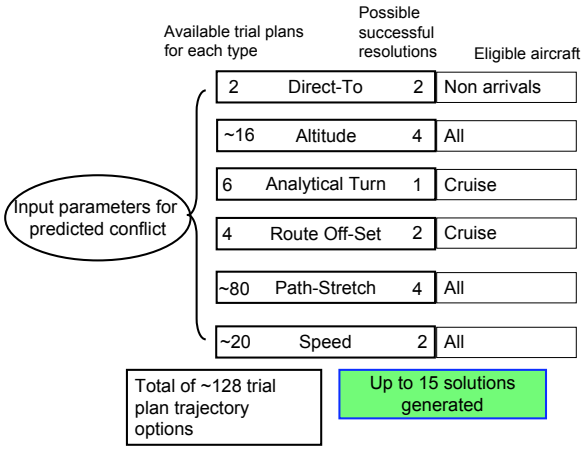


Fig. 5 Resolution types and trial trajectories generated by Resolution Generator.

5 Arrival Manager

The Arrival Manager performs arrival scheduling, sequencing and conflict resolution. It generates a four-dimensional descent trajectory for each arrival using an available trial trajectory engine. The Arrival Manager has been designed to operate either stand-alone or in conjunction with a separate scheduling and metering system such as the Traffic Management Advisor (TMA), which is widely used to control arrival traffic at U.S. airports [11].

Arrival management requires that consecutive arrivals crossing an arrival fix be separated by a minimum time interval, ΔT_{min}. A predicted violation of ΔT_{min} defines a new type of conflict,

referred to as a sequencing conflict. A sequencing conflict can occur only at a spatial location such as an arrival fix or merge point. The quantity, ΔT_{min}, referred to as an in-trail spacing constraint, is assumed specified by an arrival metering system such as TMA or by a controller. In general, ΔT_{min} is a variable that may depend on the types of aircraft in a sequence as well as on other factors. Its smallest usable value is around one minute. Smaller values are likely to produce a loss-of-separation conflict, rendering it superfluous. Thus, arrival aircraft may be involved in either or both conventional loss-of-separation conflicts and sequencing conflicts simultaneously. The Arrival Manager resolves both types of conflicts while also attempting to meet an absolute arrival fix crossing time, if one is specified by TMA. When operated in the standalone mode, the Arrival Manager generates trajectories that minimize delay to cross the arrival fixes subject only to the in-trail spacing constraint ΔT_{min}. In this mode traffic flowing to different arrival fixes is treated as independent and uncoordinated streams, sometimes called free flow in current operational practice. Free flow may be appropriate in moderate traffic conditions. When arrival traffic at an airport is approaching landing capacity, however, the Arrival Manager and TMA should be operated as a coupled system in order to ensure that the flow at arrival fixes is properly coordinated. In the coupled mode, TMA provides arrival times to the Arrival Manager, which uses its meet-time and conflict-resolution algorithms to generate conflict-free descent trajectories. The Arrival Manager also enforces an in-trail spacing constraint in the coupled mode using either a minimum default value or a value specified by TMA. In conflicts between the two constraints, the Arrival Manager gives priority to in-trail over arrival time constraints.

Similar to TMA, the Arrival Manager uses a Freeze Horizon, nominally set to 20 minutes, to determine when an aircraft first becomes eligible for arrival sequencing. Instead or in addition, a distance-based freeze horizon, typically located on a 150-nmi arc distance from the arrival fix for jet aircraft, can also be specified. For all aircraft proceeding to an arrival fix, the Trial Trajectory Engine, configured as a predictor, provides the Arrival Manager periodically (at least once per minute) with updated values of estimated times of arrival (ETAs) to the arrival fix. When the difference between the ETA of an aircraft and current time becomes less than the Freeze Horizon for the first time, the aircraft has crossed the Freeze Horizon and becomes eligible for arrival sequencing.

The Arrival Manager differentiates between aircraft that were sequenced conflict-free at an earlier time, such as aircraft A1, A2, and A3 in Fig. 6, and those, such as aircraft A4 and A5, that have crossed the Freeze Horizon at the current scheduling time and have therefore become eligible for sequencing. Previously sequenced aircraft have frozen schedules (blue color), referred to as Scheduled Times of Arrival (STAs). Their arrival trajectories are usually not changed when new arrival aircraft cross the Freeze Horizon. The strategy of sequencing arrival aircraft only once after they have crossed the Freeze Horizon ensures stability of trajectories and fairness in the sequencing process. The ETAs used in the scheduling process at freeze time are referred to as the Original ETAs (OETA). They are computed close to the time the aircraft cross the Freeze Horizon and determine the first-come-first-served sequence order. This process is similar to that used in TMA [11].

A special case arises if at freeze time the Arrival Manager finds that an arrival is in conflict with a weather cell on its current route to the arrival fix. In that case the Arrival Manager first requests the Weather Avoidance Algorithm, described in a later section, to reroute that arrival around the weather cell. Because the increased delay generated by the re-route is likely to yield an ETA that places the aircraft temporarily outside of the Freeze Horizon, the Arrival Manager defers freezing that aircraft until it again crosses the Freeze Horizon. It should be noted that the OETA at freeze time is reset to a value that includes the re-route delay.

The basic operation performed by the Arrival Manager is to merge all new aircraft that

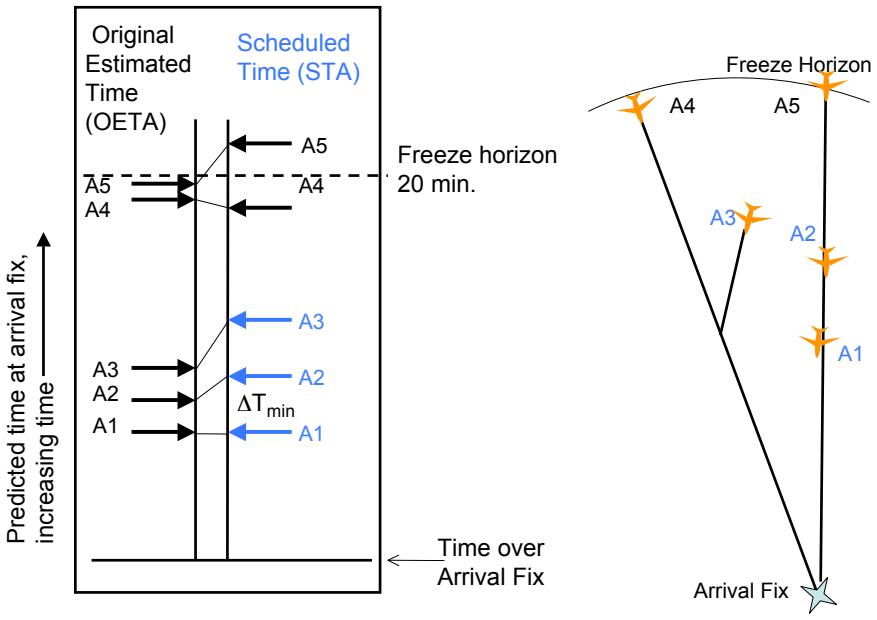


Fig. 6 Scheduling process for new Arrivals A4 and A5.

have crossed the Freeze Horizon at the current scheduling epoch into the set of frozen aircraft without causing separation or sequencing conflicts relative to each other or with frozen aircraft. The first step in this process is to find available time slots for them at the arrival fix. The OETAs of the new aircraft determine the order in which they are merged with the sequence of frozen aircraft. Next, the merged sequence is examined to identify sequencing conflicts between frozen and new aircraft and between pairs of new aircraft.

The Arrival Manager then proceeds to resolve predicted separation conflicts starting with the aircraft that has the smallest OETA (is closest in time to the arrival fix) and continuing until the last of the new conflicts has been resolved. If the Arrival Manager succeeds in finding a trajectory that resolves both types of conflicts, it designates that aircraft as frozen and scheduled. Thus, after this process has been completed for all new aircraft, the STA time line shows only frozen aircraft. The Arrival Manager then waits for real time to advance to the next scheduling epoch. When that time is reached the Arrival Manager repeats the process for all new aircraft that have crossed the Freeze Horizon. The Arrival Manager performs this process for traffic flowing to all arrival fixes at an airport. Moreover, if jet, turboprop and piston aircraft are assigned separate crossing altitudes, as is often the case, the Arrival Manager processes this traffic as independent streams.

Several rules and procedures direct the process of generating trajectories for each new aircraft. For either sequencing or combined sequencing and loss-of-separation conflicts, the Arrival Manager first determines the earliest time (STAmin) for the aircraft to be scheduled that meets the required separation time constraint relative to the STAs of frozen aircraft and is equal to or greater than the earliest time an aircraft can reach the arrival fix. The Trial Trajectory Engine is called to calculate the earliest time by using the aircraft's fastest descent speed profile. STAmin can be earlier or later than the OETA, although more often it will be later if frozen aircraft are close by and in front of an unscheduled aircraft. This is typical during an arrival traffic rush. Next, the Arrival Manager calls the Trial Trajectory Engine to compute a meet-time trajectory that achieves the STAmin and then the Trial Conflict Probe to check for conflicts. Speed profile changes are tried first and if those fail, the elliptic path stretch algorithm is used. If a trial tra-

jectory meets the time and is conflict-free relative to trajectories of all frozen aircraft, it is frozen and becomes the arrival trajectory that can be uplinked to the aircraft. The Arrival Manager then proceeds to process the next-in-line new aircraft. On the other hand, if the trial plan trajectory that achieves STAmin is found to produce a loss-of-separation conflict with a frozen aircraft, the resolution logic generates different trial trajectories at STAmin, using the elliptic path stretch algorithm. If none are conflict-free at STAmin, the Arrival Manager increases the delay by a small increment and repeats the previous process. The first delay increment that yields a trial trajectory free of both types of conflicts terminates the process and results in acceptance of the trajectory. Here it is important to note that because of the sequential procedure for processing and then freezing one new aircraft after another, conflicts between an aircraft currently being processed and new aircraft not yet processed are temporarily ignored. These induced secondary conflicts are dealt with when the next unscheduled aircraft is processed.

New arrivals with only loss-of-separation conflicts are handled slightly differently. If such a conflict occurs between aircraft in the same stream class (i.e. aircraft assigned the same crossing altitude), the Arrival Manager asks the Resolution Generator to find resolutions appropriate for arrivals and then checks each conflict-free resolution for induced sequencing conflicts with any frozen aircraft in the same stream class. Thus, a pure loss-of-separation conflict may transform into a dual conflict. The Arrival Manager treats the most difficult such conflicts in the same way as sequencing conflicts.

The procedures described are illustrated in Fig. 6, which shows both a time line plot and horizontal view of traffic converging on an arrival fix. Two new arrivals, A4 and A5, have crossed the Freeze Horizon and have become eligible for scheduling. The OETAs indicate they are not in sequencing conflict with the frozen aircraft A1-A3 but are in sequencing conflict with each other. By computing the time difference between the A5 OETA and the A4 STA, the Arrival Manager determines that a sufficient time gap is available to try increasing the speed of A5 in order to resolve or partly resolve the sequencing conflict between A4 and A5. By using the Trial Trajectory Engine and Trial Conflict Probe, the Arrival Manager determines STAmin, the earliest time A4 can cross the Arrival Fix within the speed envelope of the aircraft and without creating conflicts with frozen aircraft. This time becomes the STA for A4 as shown in the time line of Fig. 6. The Arrival Manager then calculates the STA for A5 by adding ΔT_{min} to the STA of A4. Next, the Arrival Manager asks the Resolution Generator to find a conflict-free trajectory for A5 at the specified STA. In this case a reduced-speed descent profile that meets the required conditions is found. After the scheduling process for all new arrivals is completed, the Arrival Manager waits for time to advance to the next scheduling update cycle.

The Arrival Manager has special functions to handle aircraft that appear in the arrival airspace well inside the Freeze Horizon. Such aircraft, called popups, are, of course, undesirable because they are generally difficult to merge into the frozen aircraft stream, but they cannot always be prevented. Several unplanned events contribute to their occurrence. For example, previously scheduled and frozen aircraft that have developed large trajectory tracking errors that produce conflicts with other on-time aircraft are classified as a type of popup. Onboard emergencies and weather diversions are other reasons for their occurrence. To handle them, the Arrival Manager begins by using a slot-finder function that scans the scheduled times of frozen aircraft to locate the nearest available free time slot. An acceptable time slot must have at least a time gap of twice ΔT_{min} between frozen aircraft. After locating the earliest open time slot, the Arrival Manager requests the Resolution Generator to search for a feasible trajectory for that slot. If none is found the Arrival Manger tries the next available slot. Several such iterations may be necessary before a solution is found. These techniques are illustrated in Figs. 7 and 8 for an example popup A6. In the time line of Fig. 7, the ETA of A6 falls between A1 and A2 in the frozen schedule. The

frozen schedules of A1, A2 and A3 are spaced without excess time gaps, leaving no room for another aircraft to be inserted. The slot finder inserts A6 between A3 and A4, where a gap of sufficient length is located, resulting in a delay of approximately 2.5 minutes. With a trial STA known, the Arrival Manager requests a trajectory solution from the Resolution Generator. The first solution obtained matches the specified STA, but causes a conflict with A3, as illustrated in Fig. 8. Iteration of the elliptic path stretch algorithm results in a modified path stretch that resolves the conflict while retaining the specified STA.

The method for handling popups described above assumes the Arrival Manager is operating in the standalone mode. If it is connected to TMA, then the Arrival Manager could start with the STA provided by TMA. However, since this STA is not guaranteed conflict free, the Arrival Manager still may have to change it in order to avoid conflicts. Thus, the Arrival Manager's integrated slot finder and conflict resolution functions are also required in the TMA-connected mode.

A set of rules has been developed to determine the order of resolution of a list of conflicts. The highest-level rule requires that arrival vs. arrival conflicts within a stream class have priority over all other conflict types in the resolution process. Furthermore, the resolution process starts by resolving conflicts at the fix having the highest arrival traffic count at the airport with the most traffic in the Center. For example, at the Cleveland Center, the SE arrival fix for the stream class of jets to Detroit (DTW) often experiences the highest traffic demand and is therefore selected as the starting point for the resolution process.

While resolving the arrival conflicts in this stream class, secondary conflicts may be created with out-of-stream class traffic and with nonarrivals. These secondaries are treated as new conflicts that will be resolved in later steps. Next, conflicts in the turboprop stream class to the same arrival fix are resolved, followed by conflicts in the prop stream class to this fix. The same procedure is repeated for all remaining arrival fixes at this airport, where the order in which the fixes are selected is determined by their traffic-demand ranking. At each stage in this process secondary conflicts with aircraft sequenced and deconflicted in previous steps are not permitted, while secondaries generated with traffic in stream classes not yet processed, and with overflights, are temporarily deferred. This process is repeated for all other airports in the Center, where the order of airports is determined by their rank based on traffic demand. At each scheduling epoch, the last step is to resolve the non-arrival conflicts. The conflicts resolved in the last step include the original set of non-arrival conflicts as well as deferred secondary conflicts between arrivals and nonarrivals that were created in the process of sequencing and deconfliction of the arrival traffic.

The order of processing arrivals to airports and arrival fixes in a Center, as described above, has the objective of reducing the overall complexity of scheduling and conflict resolution problems. By starting the scheduling and deconfliction of arrivals at the busiest airports and arrival fixes, the method ensures that any out-of-stream-class deferred conflicts generated by the process will involve arrivals to less busy fixes or airports where they are easier to resolve. For example, if scheduling a jet to a high-traffic arrival fix creates a deferred conflict with a turboprop at the same arrival fix, the deferred conflict is more likely easier to resolve, because traffic at turboprop arrival fixes is usually not as heavy as traffic at jet stream class fixes. In general, it is preferable to defer conflicts into lower traffic streams than into higher traffic streams. For the same reason, arrivals to busier airports are processed before arrivals to less busy airports.

6 Weather Avoidance

Convective weather, such as thunderstorms, poses a danger to aviation operations. Pilots will choose their flight plans to avoid flying through them if their location is known prior to departure. However, during the summer in certain areas of U.S. airspace, convective weather cells can develop rapidly, often without much warning. The air traffic control system receives frequent up-

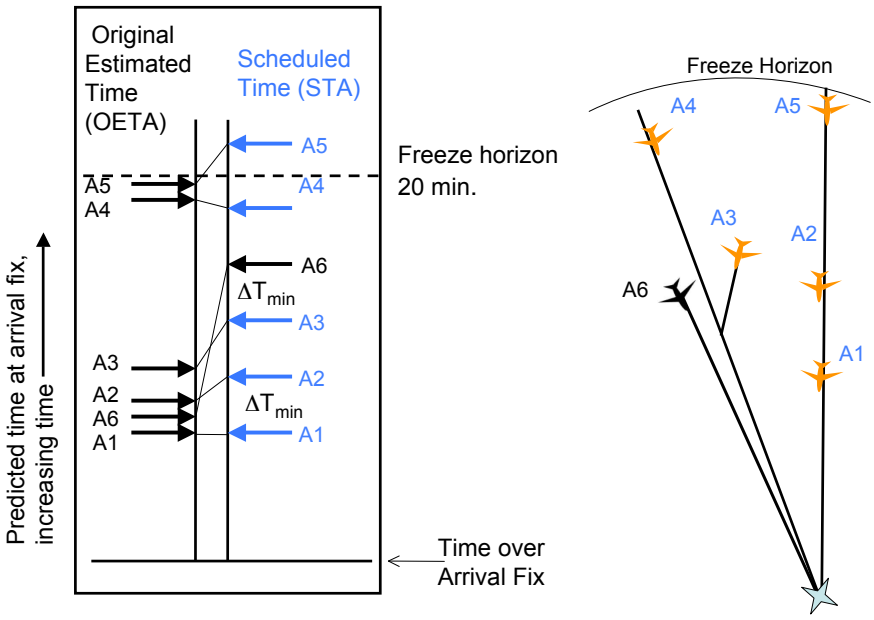


Fig. 7 Finding a slot and an STA for popup A6.

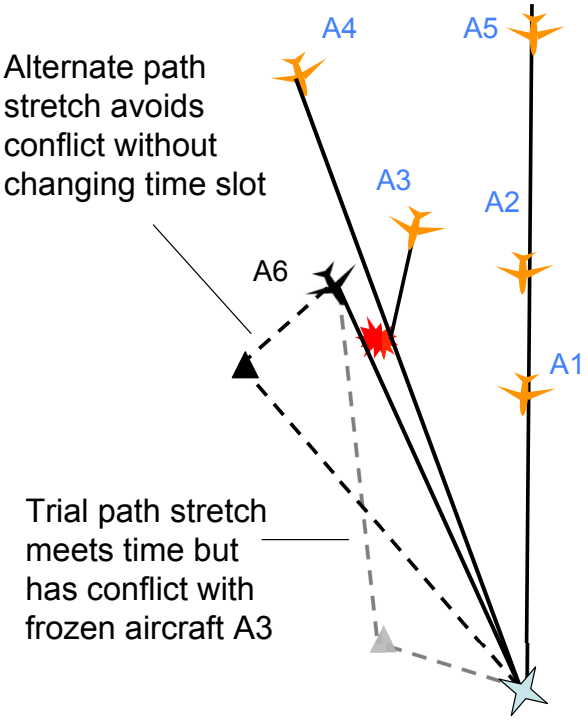


Fig. 8 Iterating on elliptic path stretch to hit time slot conflict free.

dates of convective weather activity from the National Weather Service. Regions of convective weather are displayed in air traffic control facilities on weather monitors and are also shown on controllers' traffic displays. Controllers use the location of the cells and their predicted movement to reroute traffic around the cells. This activity contributes significantly to controller workload, especially when large numbers of flights are heading into impacted regions. The controller not only has to formulate a reroute plan for each impacted flight, but must do so without creating conflicts with surrounding traffic. Therefore, weather-cell and conflict avoidance are viewed as inseparable problems to controllers. Similarly, algorithms must also be designed to solve each problem without introducing a new problem or to solve both problems if they occur simultaneously.

The solution presented here is designed to avoid weather cells that are relatively close to the current position of an aircraft, in the time range of 4 to 30 minutes to penetration. For greater times to penetration, a strategic weather reroute system is assumed to have responsibility for rerouting traffic around weather. The 4 to 30-minute time range is similar to that used in solving loss-

of-separation conflicts, allowing reuse of some of the resolution techniques developed for resolving aircraft vs. aircraft conflicts. Only horizontal maneuvers for avoiding weather cells are presented here. Although step climbs are used in current operations by pilots to fly over the tops of cells when the cells are below the ceiling of an aircraft, they are not included in the current design. Such step climbs could be added to the design if knowledge of cloud tops becomes available.

A convection cell is assumed to be an irregularly shaped region of airspace as shown in Fig. 9. While several cells separated by narrow corridors can occur, it is assumed these cells have been combined into a single larger cell before it is sent to the algorithm. A bit map is typically used to specify the area of the weather cell. Initially, an algorithm was developed that generated avoidance trajectories using the bit-map data in conjunction with separate weather-penetration-detection software. However, a more efficient approach consists of first fitting a polygon to the boundary of the bit-map weather cell. Then, an algorithm is designed to find a path around the polygon.

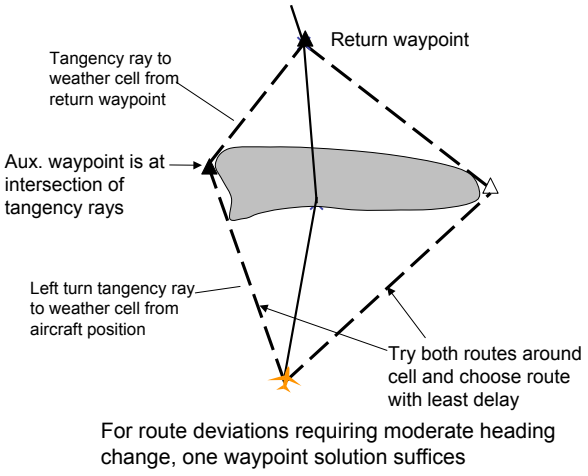


Fig. 9 Basic weather cell avoidance algorithm.

An algorithm based on this approach is illustrated in Fig. 9 for a weather cell that can be avoided by a trajectory using a single auxiliary waypoint. The solution obtained is structurally similar to a path-stretch maneuver used in conflict resolution. The process starts by computing a ray tangent to the weather-cell polygon emanating from the current position of the aircraft. Open source software for computing such tangent rays is incorporated in the algorithm. A second ray emanating from a specified return waypoint in the backward direction and tangent to the polygon is also computed. The location of the auxiliary waypoint is defined as the intersection of the two tangent rays. The solution of two simultaneous linear equations determines the coordinates of the waypoint location. The path obtained is the shortest around the weather-cell polygon, given that only a single auxiliary waypoint can be used in the construction of the path. A second path in the opposite turn direction is also computed, and then the shorter of the two paths is chosen as the avoidance trajectory.

The auxiliary- and return-waypoint coordinates are sent to the Trial Trajectory Engine and Trial Conflict Probe to generate the avoidance trajectory and check it for induced conflicts. If a conflict is detected within the resolution time horizon (typically 8 minutes for non-arrivals and 20 minutes for arrivals) as illustrated in Fig. 10, the Weather-Avoidance Algorithm requests the Resolution Generator to resolve it. The request includes constraints on the preference order and type of resolution that is acceptable for weather induced conflicts. Altitude resolutions are preferred, because they preserve the existing weather-avoidance path. If they are unsuccessful, a path-stretch maneuver is tried where the return waypoint for the path stretch is chosen to be the auxiliary waypoint for the weather-avoidance maneuver. In that case the composite trajectory contains two auxiliary waypoints, one for weather and one for conflict avoidance. It should be mentioned that the resolver also ensures that the path stretch maneuver does not reenter the weather-cell area.

The trajectories obtained from this procedure are adequate for those weather-cell encounters that require heading changes at the auxiliary waypoint of less than 90 degrees. When this limit is exceeded, or if the tangent rays do not intersect, a procedure requiring two waypoints is used, as illustrated in Fig. 11. While the shape

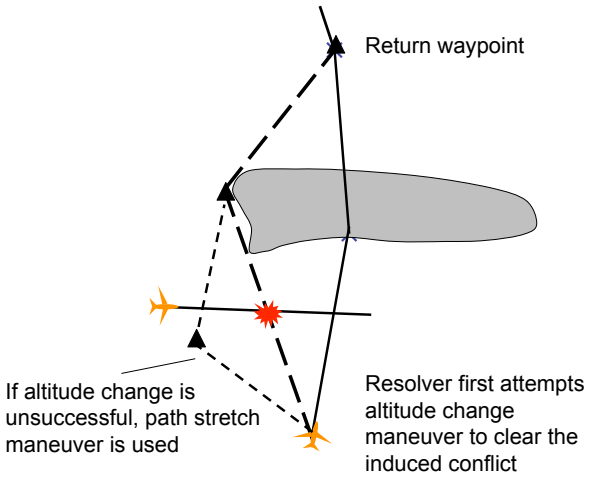


Fig. 10 Resolution of weather avoidance induced conflict.

of the weather cell as well as the encounter scenario shown in Fig. 11 appear extreme, they were selected to demonstrate the ability of the two-waypoint approach to handle unusual scenarios. In a recent human-in-the-loop simulation that included weather, controllers occasionally used multiple waypoints and large heading changes to reroute traffic around wide areas of convective weather cells, resulting in paths similar to the one in Fig. 11. These observations of controller techniques suggested the need for handling extreme situations and motivated the two-waypoint approach.

The procedure for the two-waypoint resolution begins the same way as for the single waypoint case. It uses the tangent-ray calculation software to determine two tangent rays to the weather-cell polygon, one emanating from the current aircraft position and the other emanating from a specified return waypoint in the backward direction. Then a trial location is chosen for the first auxiliary waypoint a short distance past the point of tangency of the ray emanating from the aircraft position. Next, a tangency ray emanating from the trial first waypoint location is calculated. The coordinates of the trial second waypoint are fixed by the intersection of two tangency rays, one emanating from the first and the second from the return waypoint. The coordinates as well as the lengths of two segments are obtained as the solution of two simultaneous linear equations defined by the two rays. If the rays intercept, the total length of the resolution path is calculated by summing the known lengths of the three path segments. The first trial waypoint is now moved an incremental distance (10 nmi) further along the tangency ray and the previous calculations are repeated to determine a new value for the path length. Incremental iteration of the first waypoint location is repeated until the minimum value of path length is found. Thus, the solution obtained by this iteration process determines the smallest delay weather avoidance path defined by three linear path segments and two auxiliary waypoints. It is possible that the initial location of the first waypoint may not yield a tangency line that intercepts the third tangent line. In such a situation, the trial first waypoint location is moved in increments until an interception point is found. Before accepting the two-waypoint solution, the algorithm checks whether the length of the middle segment exceeds a minimum value (20 nmi). If it does not, the one-waypoint solution described earlier is computed to replace the two-waypoint solution.

7 Web-Accessible Documentation and Simulation Results

The Autoresolver, comprising the Resolution Generator, Arrival Manager and Weather-Avoidance Algorithm, has been implemented in software and written in Java. The Autoresolver is linked to a fast-time simulation in an ongoing effort to improve the algorithms and evaluate their performance. In addition, a web-based data-analysis tool was developed for analyzing simulation results. The tool saves the output of simulation runs and provides users access to various statistical measures of algorithm performance as well as lists of resolved conflicts, trial trajectory iterations, and graphs of resolution trajectories [12]. The website also contains an interactive software design document that allows users to examine logical decision variables used in the resolution process and follow branches in the decision tree that lead to various resolution types.

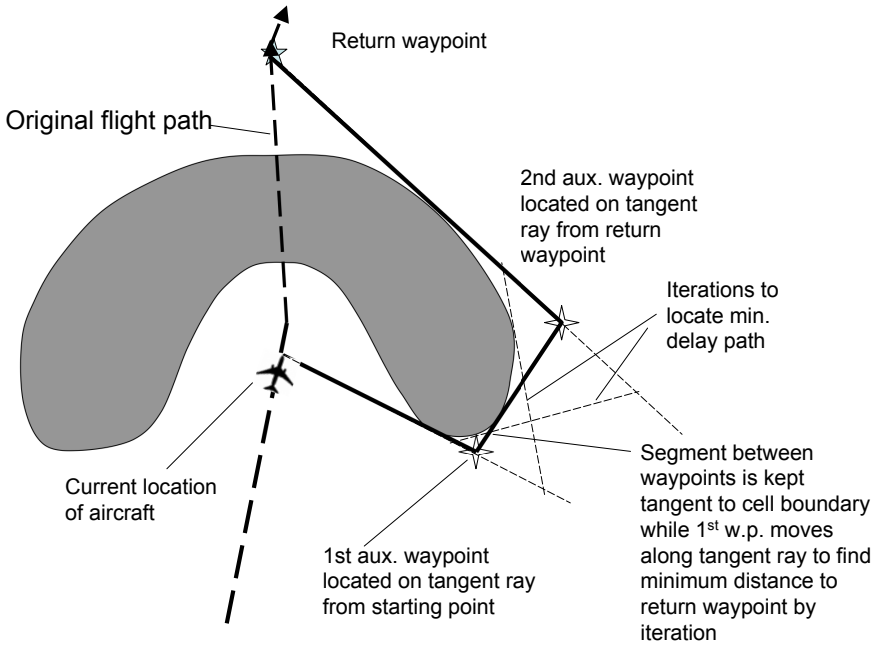


Fig. 11 Using two auxillary waypoints for large turn angles (Finding a way out of a box-canyon weather cell).

The online document identifies more than 20 resolution types used by the Autoresolver, including certain types not described in this paper.

The Advanced Concepts Evaluation System (ACES) was chosen as the fast-time simulation environment for development and testing of the Autoresolver [13]. ACES uses as input flight plans of traffic recorded over a 24-hour period on a selected date. All flights that operate in US airspace on that date are included in the recorded flight plans. The simulation uses these flight plans together with aircraft performance models to generate pseudo radar tracks. The Autoresolver, upon receiving these tracks and flight plans, issues resolution trajectories to aircraft in conflict, which respond to the trajectories in a way similar to the way live traffic would. Furthermore, ACES has the ability to scale up the originally recorded traffic level by replicating flight plans a specified number of times. The performance of the Autoresolver, including all three of its resolution and problem solving functions, could thus be evaluated at traffic levels expected in future years.

Fig. 12 shows simulation results of Autoresolver performance for the Cleveland Center at three traffic levels, corresponding to current level, 1x, and two future levels, 2x and 3x. Arrivals to all major airports in the Center are included. The performance measure shown is the average delay generated by the resolution maneuvers. All detected nonarrival conflicts, defined as those not involving two arrivals destined to the same arrival fix, were resolved. The delays for the three traffic levels ranged from a few seconds at 1x to about 10 seconds at 3x. These delays are reduced by a factor of three compared to those obtained in [2] for comparable traffic levels. The reductions are attributable to optimization of resolutions performed by the Resolution Generator. Resolution delays grow with increasing traffic levels as one would expect, but they remain relatively low even at 3x.

The resolution of arrival conflicts was limited to those that could be sequenced and scheduled with less than 5 minutes of delay. This restriction was necessary at the 2x and 3x levels because these levels exceeded the capacity of the four arrival fixes used in current operations at the Detroit airport. Without a limit on arrival traffic, path-stretch delays would become unrealistically long and holding would be necessary. Simula-

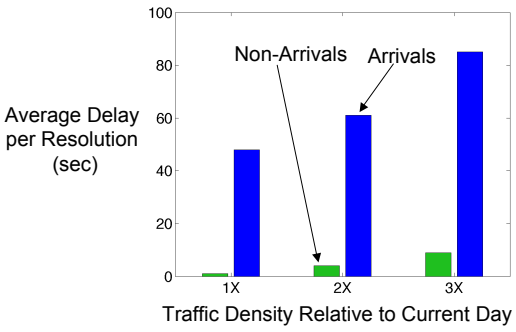


Fig. 12 Resolution delays at 1X, 2X and 3X traffic levels, Cleveland Center. Arrival delays for Detroit airport (DTW).

tion runs conducted without the 5-minute limit found that the Arrival Manager was still able to sequence the higher traffic levels conflict-free, although at the cost of very large delays. In practice, the traffic flow management systems would limit the rate of arrival traffic upstream well before excessive delays could build up at an airport. Average arrival delays of 48 seconds at the 1x traffic level are found to be about 50% less than the delays obtained for the same condition in the earlier study [2]. These delay reductions can be attributed to the Arrival Manager's improved handling of arrival traffic. A direct comparison of performance could be made only at the 1x level, because at this level the 5-minute delay limit was not exceeded by any conflict and thus could not distort the delay comparison.

Table 1 gives the percentage of resolutions contributed by each of eleven resolution types for the three traffic levels. The table also lists the total number of simulated flights and conflicts resolved at each traffic level. Path stretches are the most frequently used maneuver, contributing approximately 38% to the total. Next in order are step and temporary altitude maneuvers, followed by Direct-To. Together, the top four make up 80% of the total resolution maneuvers. Nevertheless, the less frequently used maneuvers such as speed and descent speed profile are critically important in the arrival management process, where they play a significant role in resolving sequencing conflicts. While the offset was not often the selected maneuver, it is an important alternative maneuver available for controllers or pilots to select instead of path stretches. The data in the table also show that the percentage of each maneuver type used in resolutions changes little with traffic level. Though not included in the table, the percentage of conflicts with multiple resolution options was found to be 99% for en route and 84% for arrival conflicts. This percentage was only slightly affected by changes in traffic density.

Table 1 Resolution statistics for three traffic level, Cleveland Center, ACES fast time simulation

Traffic level	1x	2x	3x
No. of flights in 24 hours	7434	15385	26833
No. of conflicts	1040	3887	10865
Direct-to, % of total	12	11	11
Analytical turn, %	.8	.4	.6
Path stretch, %	37	39	39
Route offset, %	1.8	2.5	2.7
Step altitude, %	18	18	23
Temp. altitude, climb, %	12	15	13
Temp. altitude, arrival, %	2.7	2.9	2.5
Cruise speed, %	7	7	4
Descent speed profile, %	8	6	4
Reduced cruise altitude, %	.48	.13	.1
Path stretch plus speed, %	.2	.23	.7

The software for the algorithm allows users to favor a specified class of maneuvers from the set of alternatives by means of a preference parameter. This parameter could, for example, be set to favor horizontal over vertical maneuvers or offsets over path stretches. A consequence of giving preference to a maneuver is that a non-minimum delay time maneuver will be selected.

NASA recently conducted a real-time simulation of a NextGen concept where controllers used the Autoresolver as a decision support tool [14]. This simulation is the latest in a series of human-in-the-loop simulations that have used the Autoresolver to investigate several operational con-

cepts for automation-assisted separation assurance. Results of earlier simulations are described in [7]. Controllers have generally found the resolutions offered by the Autoresolver acceptable and, by using it as a decision support tool, were able to handle substantially increased traffic levels.

8 Evolutionary Path to Automated Separation Assurance

Major changes in air traffic control operations have historically evolved gradually, often over decades. It is therefore important to consider evolutionary steps that could lead toward higher levels of automation over a period of time. The system architecture and the algorithms outlined in this paper should be considered as the final step in a future air traffic system that will include a high level of automation. Both the system architecture and the algorithms described in the paper lend themselves to operational implementation in evolutionary steps. The two main options for stepwise implementation are short-range conflict resolution and strategic conflict resolution. As described in [4], the first option, which is referred to as TSAFE, would be a paradigm-shifting step in that it would largely remove the controller from responsibility for both detection and resolution of short-range conflicts. The two supporting technologies required for implementing this option are the data link built into Mode S and the onboard systems required for TCAS level 2. These technologies are already in operational use and could be adapted for this application with relatively minor modifications to systems onboard aircraft. In addition, the short-range detection and resolution algorithm would also have to be implemented in the ground system. Alternatively, it may also be possible to implement TSAFE or similar technologies as an airborne separation-assurance system that is independent of systems on the ground. The second option would be to implement strategic conflict resolution using an algorithm such as the Autoresolver described in this paper. The Autoresolver would have to be integrated into the ground system where it would initially be used as a decision support tool by controllers. Controllers would issue the resolution trajectories to equipped aircraft primarily via data link. This option is critically dependent on the installation of a ground-air data link in a large proportion of aircraft. Such a data link is under development and is expected to become operational in a few years. Once a data link becomes widely available, automated up linking of trajectories could be implemented first in high-altitude en-route airspace and then could be followed later by implementation in transition airspace. After both the short-range and strategic resolution functions have been successfully used over a period of time, these functions could be operated as an integrated system, thereby achieving a level of automated separation assurance envisioned for an advanced version of NextGen.

9 Concluding Remarks

This paper describes a set of algorithms that solves three types of problems encountered in automated separation-assurance. Because the three types of problems, defined as conflict resolution, arrival management and weather avoidance, can arise separately or in any combination, the design described in the paper unifies and integrates the separate algorithms for solving these three problems. The algorithm for conflict resolution improves upon an earlier version by generating different types of resolutions for each conflict and then ranking the successful resolutions in time-delay order. This design change responds to requests for alternative solutions from experimental users of the algorithm. It has also improved efficiency by significantly reducing the time delay of resolutions. The Arrival Manager solves the combined problems of sequencing, spacing, and conflict resolution of arrival traffic. It can be operated either in a stand-alone mode or in a mode integrated with an arrival-traffic metering system. A weather-avoidance algorithm reroutes aircraft around convection cells using up to two auxiliary waypoints. It also ensures that conflicts are resolved and arrivals sequenced without penetrating weather cells. The maximum benefit

from the trajectories generated by the algorithms will be achieved by using a data link to uplink them to the target aircraft. The paper suggests a roadmap for deploying automated separation- assurance functions in evolutionary steps, leading eventually to a system with increased efficiency and capacity, wherein controllers have substantially less responsibility for separation- assurance than in today's operations.

10 Copyright Statement

References

[1] Erzberger H. Transforming the NAS: The next generation air traffic control system. *Proc 24th Iternational Congress of the Aeronautical Sciences*, Yokohama, Japan, 2004.

[2] Erzberger H. Automated conflict resolution for air traffic control. *Proc 25th Iternational Congress of the Aeronautical Sciences*, 2006.

[3] Farley T and Erzberger H. Fast time air traffic simulation of a conflict resolution algorithm under high air traffic demand. *Proc UAS/Europe ATM R&D Seminar*, Barcelona, Spain, 2007.

[4] Erzberger H and Here K. Algorithm and operational concept for resolving short range conflicts. *Proc 26th International Congress of the Aeronautical Sciences*, Anchorage, Alaska, 2008.

[5] Andrews J, Erzberger H, and Welch J. Safety analysis for advanced separation concepts. *Air Traffic Control Quarterly*, Vol. 14, No 1, 2005.

[6] Kupfer M, Farley T, Chu Y, and Erzberger H. Automated conflict resolution-a simulation-based sensitivity study of airspace and demand. *Proc 26th International Congress of the Aeronautical Sciences*, Anchorage, Alaska, 2008.

[7] Prevot T, Homola J, Mercer J, Mainini M, and Cabrall C. Initial evaluation of air/ground operations with ground-based automated separation assurance. *Proc 8th USA/Europe ATM R&D Seminar*, Napa, California, 2009.

[8] Erzberger H, Paielli R, Isaacson D, and Eshow M. Conflict prediction and resolution in the presence of prediction errors. *Proc 1st USA/Europe ATM R&D Seminar*, Saclay, France, 1997.

[9] McNally D, Engelland S, Bach R, Chan W, Brasil C, Gong C, Frey J, and Vincent D. Operational evaluation of the direct-to controller tool. *Proc 4th USA/Europe ATM R&D Seminar*, Santa Fe, New Mexico, 2001.

[10] Bach R, Farrell C, and Erzberger H. An algorithm for level-aircraft conflict resolution. *NASA/CR-2009-214573*, 2009.

[11] Swenson H, Hoang T, and Engelland S. Design and operational evaluation of the Traffic Management Advisor at the Fort Worth Air Route Traffic Control Center. *Proc 1st USA/Europe ATM R&D Seminar*, Saclay, France, 1997.

[12] Website for performance statistics, resolution trajectories and sofware documentation of Autoresolver algorithm described in this paper and in references 2, 3 and 6. http://airtrafficconflictresolutions.arc.nasa.gov.

[13] Meyn L, Windhorst R, Roth K, Drei D. V, Kubat G, Manikonda V, Roney S, Hunter G, and Couluris G. Build 4 of the airspace concepts evaluation system. *Proc AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2006.

[14] Mercer J, Prevot T, Brasil C, Mainini M, Kupfer M, and Smith N. An integrated tool suite for en route radar controllers in NextGen. *Proc 27th International Congress of the Aeronautical Sciences*, Nice, France, 2010.