

Coordination Between Multiple Ground-Based Separation Assurance Agents

Todd A. Lauderdale* and Tony Wang†

NASA Ames Research Center, Moffett Field, CA, 94035

A ground-based prototype for automated separation assurance is evaluated where the airspace is divided into multiple regions controlled by independent agents. To ensure that these distributed, independent agents work together to provide seamless, safe, and efficient operations, coordination amongst these agents is required. This paper will present a set of prioritization rules that provide this coordination for both external agents and different regional agents of the same type. Simulations of the impact of these rules are presented to highlight the coordination problem solved by the specific rules. The final result of these coordination schemes is a computationally-efficient, conflict-free separation assurance capability distributed among multiple agents.

I. Introduction

SPATIAL separation between aircraft in the current air-transportation system is maintained in a distributed, ground-based fashion with multiple regional air-traffic controllers providing separation across the national airspace system (NAS). These controllers have different areas of specialization including airport tower controllers, Terminal Radar Approach controllers, and enroute controllers. Separation assurance is handled differently in each of these regions, with the most structure and constraints for aircraft approaching the runway and the least structure for aircraft in enroute airspace. Regardless of their area of specialization, each controller is responsible for maintaining separation for all aircraft under their control, and they must ensure that hand-offs to downstream controllers are done safely.¹

The current system is divided into these multiple regions because a single person could not provide separation for the thousands of aircraft in the air at any given time. In a future fully automated separation assurance system, however, it is theoretically possible that a single agent could be responsible for maintaining separation for all aircraft, providing a global, centralized control system. Since this system could prioritize conflicts and proceed through them sequentially, it could be internally coordinated, and this would remove the need for constraints on resolutions to ensure coordination. The computation power required for a single agent to control the entire airspace might be significant, so the solution may need to be parallelized in an efficient way that still maintains safety.² A key insight in this parallelization is that, since the time scale for separation assurance is on the order of tens of minutes, a resolution in one area will not have an impact on the separation assurance problem beyond a few hundred miles. Therefore, the problem can be decomposed geographically, similar to how it is decomposed in the current system, while minimizing the synchronization and coordination required to ensure efficient solutions.

The Advanced Airspace Concept (AAC)³ seeks to automate separation assurance for enroute, center-controlled airspace. The strategic separation assurance layer of this concept is known as the AAC Autoresolver⁴ and is responsible for separation assurance from around 20 minutes down to about 2 minutes before a predicted loss of separation. The Autoresolver was initially designed and evaluated for use in a single sub-region of the NAS.⁵ Recently, it has been evaluated by treating the entire NAS as a single airspace⁶ and, also, by using multiple regional Autoresolvers with no coordination among the independent agents.⁷ Since the Autoresolver is an enroute control system, there are two types of boundaries across which the Autoresolver does not control the aircraft: terminal area boundaries and international boundaries. In all of the

*Aerospace Engineer, Aviation Systems Division, NASA Ames Research Center

†Principal Programmer, University-Affiliated Research Center, University of California, Santa Cruz

previous studies, conflicts were filtered from the results if they involved an aircraft that had been in enroute airspace for only a short time since there was no coordination between terminal areas supplying flights to the centers and the center agents. This lack of coordination resulted in aircraft entering Autoresolver control with spacing below the required standard.

Since there was no inter-agent coordination, none of these previous ground-based solutions addressed separation assurance for the entire NAS in a distributed way, and this does not seem to have been a focus of ground-based separation assurance research. Inter-agent coordination has long been an area of focus for airborne separation assurance though.⁸ This paper is intended to extend the ground-based separation research into multiple-agent, parallel solutions by presenting a set of specific coordination rules that enable a complete, parallel Autoresolver solution. In this paper the parallelization is implemented by dividing the airspace into different regions using the current Air Route Traffic Control Centers as defined by the FAA. Each region has an independent Autoresolver agent with visibility, through extended surveillance, into adjacent airspace. A coordination scheme for aircraft entering Autoresolver control from other, non-Autoresolver, agents and for conflicts with aircraft under the control of different Autoresolver instances is also presented. These coordination requirements are tailored to the separation assurance task in that they take into account location of control, prioritization of arrival flows, and prioritization of aircraft with time-based constraints. These coordination schemes are implemented in a fast-time simulation system, and results are presented to illustrate the impact of the various coordination criteria.

II. Coordination Criteria

The AAC Autoresolver is designed to handle separation assurance for enroute airspace, therefore aircraft entering this airspace from external control cross either international (Fig. 1(a)) or terminal-area (Fig. 1(b)) boundaries. Regional decomposition of the problem to facilitate parallelization results in a second type of boundary where two independent Autoresolver agents are in control of aircraft on either side of the boundary. Assuming one Autoresolver per center, these boundaries are indicated by the internal green lines in Fig. 1(a). This section will discuss the coordination requirements across these two types of boundaries.

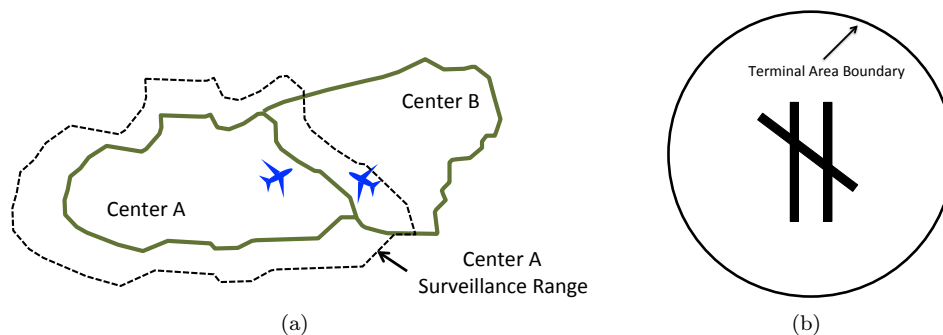


Figure 1. Schematics showing (a) the international (black line) and center regional (green line) boundaries and (b) the regional boundary between an airport and a center.

A. Non-Autoresolver to Autoresolver Boundaries

For aircraft entering Autoresolver control from an external region, an assumption is made that the Autoresolver has no control over the trajectory that the aircraft will fly prior to handoff. So, the only control Autoresolver has is to either allow or not allow the aircraft to enter its airspace. The basic rule for this hand-off is fairly simple: all trajectories must be free of conflicts with all aircraft in the Autoresolver airspace for some pre-defined time, T_{CF} . Since the Autoresolver is aware of the aircraft under its control, this coordination is enabled by allowing other agents to check their aircraft trajectory for conflicts with Autoresolver aircraft. An example where a trajectory for an externally controlled aircraft is in conflict with an Autoresolver aircraft is shown in Fig. 2(a).

So, before control of an aircraft is handed off from a non-Autoresolver region to an Autoresolver region, at the boundary between the two regions, the non-Autoresolver agent creates a trajectory for the aircraft from the current time to the boundary crossing time plus T_{CF} . The Autoresolver performs conflict detection on

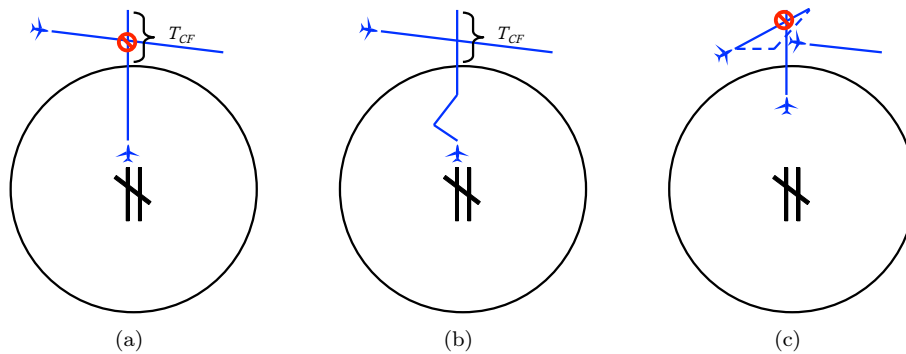


Figure 2. Schematics showing (a) a conflict between a transitioning flight and an Autoresolver flight, (b) a possible resolution that shifts the entry time of the transitioning flight, and (c) a new conflict between the frozen trajectory and an Autoresolver flight that must be resolved by the Autoresolver flight.

that trajectory, and if the trajectory is conflict-free, then the aircraft can be handed off to the Autoresolver. If the trajectory is not conflict-free, then the non-Autoresolver agent must modify the trajectory by either changing the aircraft's path and entry point into the Autoresolver region or by delaying its entry into Autoresolver control such that its new trajectory is conflict-free (Fig. 2(b)).

Once a hand-off trajectory has been agreed upon the trajectory is considered frozen by the Autoresolver in that the Autoresolver may not alter this trajectory until the aircraft crosses the boundary into the Autoresolver-controlled region and the handoff is completed. If the non-Autoresolver agent subsequently alters the trajectory for any reason the hand-off coordination process must be repeated. Any subsequent conflicts with aircraft controlled by the Autoresolver must be resolved by the controlled aircraft as depicted in Fig. 2(c), and all newly commanded trajectories in the Autoresolver airspace must be free of conflicts with all frozen trajectories.

B. Autoresolver to Autoresolver Boundaries

Coordination across boundaries where different Autoresolver agents control the aircraft on either side allows for more flexibility than when one of the regions is not controlled by an Autoresolver agent. There are three important pieces of this coordination: determining the controlling agent for each flight, determining which flights are visible to each agent, and determining which agent is responsible for each conflict.

The first rule of this coordination logic is that each Autoresolver agent can only maneuver aircraft that are under its control. It is very important for all the following coordination logic that all agents agree which agent controls each flight. There are two ways to determine the controlling agent depending on whether the aircraft is an arrival flight or not, where an arrival flight is any aircraft within a specified time (e.g. 25 minutes) of its arrival fix. For non-arrivals (Fig. 3(a)), the controlling center is determined by the current location of the aircraft, where the agent for the airspace in which the aircraft is currently located is in control of the aircraft. As shown in Fig. 3(b), for arrival flights, the controlling agent is the one responsible for the airspace in which the airport resides. In this situation, when the airport is near a center boundary, it is possible for an aircraft to be controlled by an agent in a different airspace than the one in which it currently resides. This early transfer of control to the arrival airspace agent is necessary because the arrival problem is highly constrained and difficult, and the solution must be coordinated amongst all aircraft approaching that airport.

To ensure that an agent does not make a poor decision about how to maneuver an aircraft under its control, each agent must have visibility into the adjoining region and around all aircraft that it controls. Generally, as is shown in Fig. 4(a), the surveillance range for a region extends a fixed distance beyond that region's boundary. When there is an airport near the boundary of a center, this surveillance range is extended to a certain radius around that airport to provide visibility around all early transferred aircraft (Fig. 4(b)). This requirement for extended surveillance due to airport locations is an artifact of the current airspace boundary definitions and may be removed, at least for significant airports, with redesigned airspace boundaries.

Once this visibility is established, there are many ways to ensure that the conflict resolution processes of multiple independent agents are coordinated. The following rules were developed based on the desire to

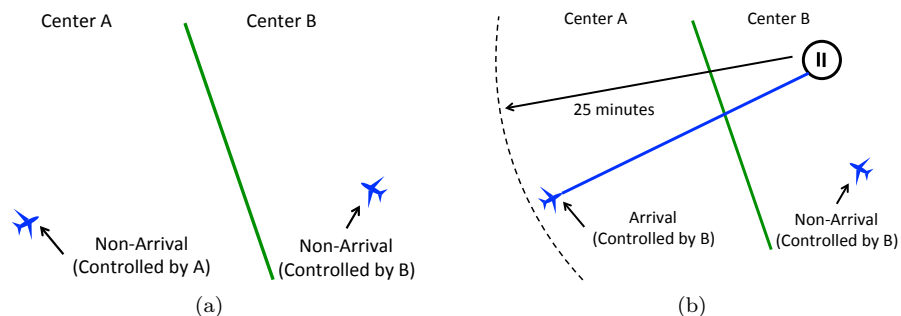


Figure 3. (a) For non-arrival flights the controlling center is the one containing the current location of the aircraft, but (b) for arrival flights, the controlling center is the center containing the airport.

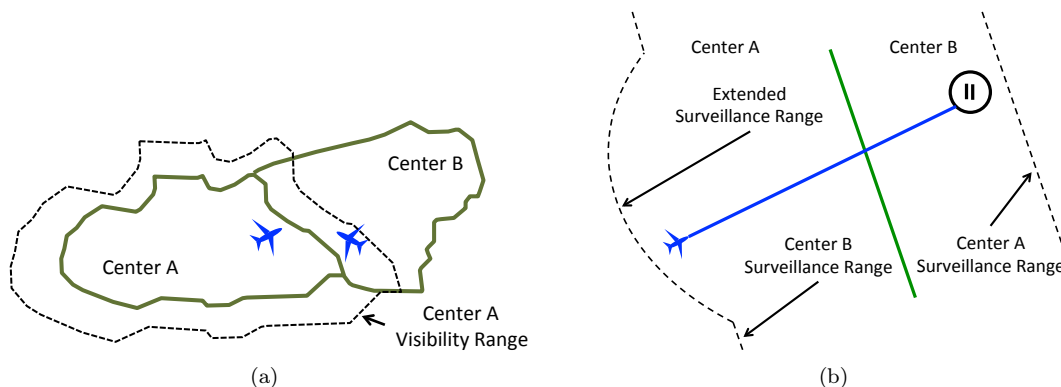


Figure 4. (a) Two adjacent regions and the surveillance range for Center A, and (b) the extended surveillance range required for airports near center boundaries.

minimize the required data sharing and communication between agents. This allows for highly independent operation of the different agents, but it comes with the added cost of occasionally requiring a less efficient maneuver since the coordination rules prescribe the aircraft to maneuver instead of choosing the most efficient resolution between the two aircraft.

The main goal of this coordination is to ensure that one, and only one, Autoresolver agent is responsible for creating resolutions for each conflict. If this holds, unsafe and inefficient situations such as those where two aircraft are maneuvered independently to resolve the same conflict or where neither aircraft is maneuvered, will not occur. If both aircraft are controlled by the same agent, no coordination is required and the coordination logic is not triggered. Given that two aircraft are controlled by different agents, the logic has been designed such that the two agents will implicitly agree which aircraft should be maneuvered instead of using active communication. This is to reduce communication and synchronization issues arising from active communication. An overview of the logic to select which aircraft to maneuver is shown in Fig. 5 and described below.

Arrivals are given preferential treatment because they are approaching a constraint at the arrival fix (Fig. 6(a)). This spatial, and often temporal, constraint at the arrival fix means that maneuvering an aircraft approaching its arrival fix can have consequences for other aircraft also approaching that fix. There are generally two types of arrival aircraft, one that has a scheduled time of arrival resulting from arrival metering and another that does not. These two types of arrivals are handled differently such that an aircraft with scheduled times of arrival has the highest priority and will not be maneuvered if at all possible. If both aircraft have a scheduled time of arrival, then the aircraft traveling to a lower priority airport is maneuvered, where the airport priority list is a constant shared with all agents. If only one of the aircraft is scheduled, then the non-scheduled aircraft must be maneuvered. If both aircraft are arrivals, but neither aircraft has a scheduled time of arrival, then the aircraft traveling to the lower priority airport is maneuvered. If only one of the aircraft is an arrival, then the non-arrival must be maneuvered (Fig. 6(b)). To accomplish this coordination implicitly, all agents must agree on the definition of an arrival aircraft, must be aware of when an aircraft is scheduled and must agree on airport priority.

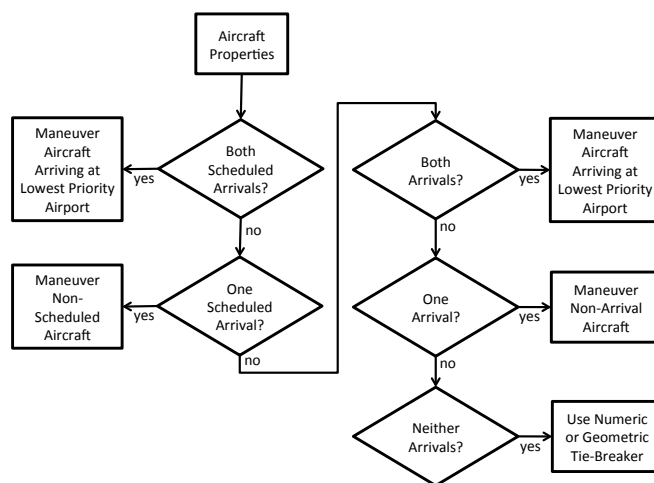


Figure 5. Flowchart showing the resolution coordination logic.

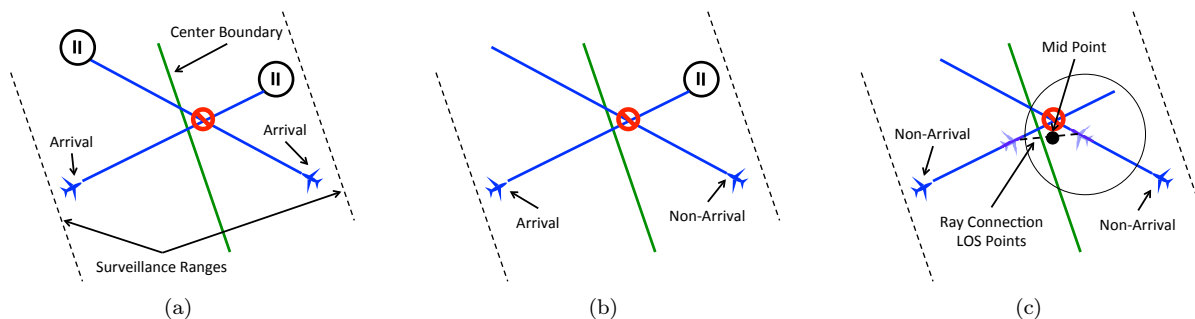


Figure 6. Schematics showing (a) a conflict between two arrival flights in conflict in two different centers, (b) a conflict between an arrival and a non-arrival flight in two different centers, and (c) a conflict between two non-arrival flights in two separate centers.

If neither aircraft is an arrival of any type, then an implicit priority must be created such that both controlling agents agree on which aircraft will be maneuvered. One way of assigning priority is to base it on the geographic location of the aircraft and the geometry of the conflict. A way to do this is to create a ray connecting the locations of the two aircraft at the predicted time of separation loss (Fig. 6(c)). The midpoint of this ray is determined and if that point lies in either of the two centers, then the center in which the midpoint lies is responsible for resolving the conflict. The final possibility is that the midpoint lies in a third center. If this is the case, a tiebreaker is used, and the aircraft with the lowest flight ID is maneuvered.

If this geometric method to determine which aircraft to maneuver is used the predicted loss points are subject to noise and prediction errors and may need to be synchronized explicitly. Also, the noise can cause switching of responsibility as a function of time for corner cases. For these reasons, a straight numerical assignment of priority may be preferable in which priority is based on some assigned number like a transponder identification code or a flight ID. This number should be known by all centers and must be equitable such that one particular aircraft, aircraft type, or airline is not burdened more frequently on average,

This set of rules was created to minimize the amount of information that must be shared between the agents, but there is some information that must be shared or that the multiple agents must agree on intrinsically. First, all agents must agree whether an aircraft is a scheduled arrival or a regular arrival. Second, they must agree on the current locations of the aircraft. And third, they must agree on the coordinates of the center boundaries. The locations of the aircraft may be subject to surveillance errors, so these locations may need to be synchronized between agents.

One limitation of using these coordination rules is that they restrict the solution space for certain conflicts from allowing both aircraft to be maneuvered to only allowing one of the aircraft to be maneuvered. This

could cause a problem if, for some reason, no resolution can be found for the selected aircraft. In practice, there were no cases in this study where the prescribed agent could not resolve the conflict. This restricted solution space does result in less efficient solutions in certain cases when only one aircraft is checked for the optimal resolution.

Another weakness of these rules is that they only ensure that solutions for aircraft in the same conflict across a regional boundary are coordinated. It is possible that two aircraft across a boundary will both be in conflict with different aircraft in their respective regions as shown in Fig. 7(a). The two resolutions required to solve the independent conflicts can then bring these two aircraft into a near term conflict (Fig. 7(b)). This is a rare problem since both conflicts need to occur at the same time in close proximity to a regional boundary. In fact, in the simulations discussed below, there was no case where two simultaneous resolutions resulted in a new conflict with time to loss of separation is less than 5 minutes. There have been infrequent cases observed in other scenarios.

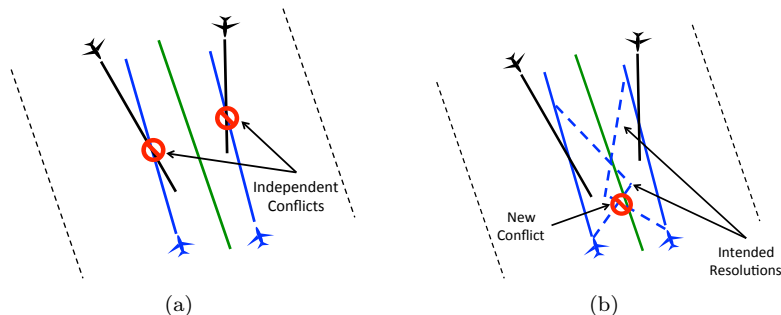


Figure 7. Schematics showing (a) two independent conflicts across a center boundary and (b) a situation where simultaneous resolutions to independent conflicts result in a near-term loss of separation.

One solution to this simultaneous-resolution coordination problem is to create a higher-level coordination agent to which all proposed resolutions are submitted. There will be a single one of these coordination agents for the entire airspace and new resolutions will be checked against pending resolutions to determine if new, near-term losses of separation are created by this solution. If a critical loss of separation is created, this resolution is rejected and the proposing airspace is notified. This feature was added to the Autoresolver and used in the subsequent simulations.

III. Simulations

The Autoresolver was initially designed and implemented in the Airspace Concept Evaluation System (ACES)⁹ using a single center (most often Cleveland Center). ACES was subsequently modified to allow for one instance of Autoresolver per center with an expanded zone of visibility across center boundaries,⁷ but no coordination was provided between agents beyond the rule that each agent could only maneuver those aircraft that were located inside its regional boundaries.

To show the impact of the coordination rules discussed in Section II, these rules have been implemented in the Autoresolver and simulated in ACES without convective weather. Different combinations of the previously discussed coordination rules were simulated using two different flight data sets, both of which contained about 10,000 nationwide flights derived from current day filed flight plans. These subsets of aircraft were selected from full-day datasets by identifying a four-hour period with the highest number of scheduled departures, and Fig. 8 shows the resulting number of aircraft flying in the simulation as a function of the simulation time for the two different datasets.

Four different coordination scenarios were simulated for each of the two datasets. In one, neither coordination between external agents and Autoresolver agents nor coordination between Autoresolver instances were used. In the second case, coordination between non-Autoresolver agents and Autoresolver agents was included, but none between independent Autoresolver agents. And in the third case both types of coordination were included. Finally, to create a baseline to determine the amount of inefficiency imposed by the implicit coordination, for the last case, a single Autoresolver agent was used for the entire airspace, so there was no coordination required between Autoresolver instances. In this case though, there was coordination between non-Autoresolver agents and the single Autoresolver instance. In all four scenarios the Arrival

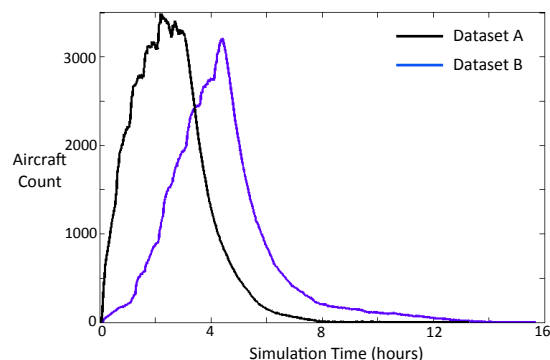


Figure 8. The number of aircraft in the simulation as a function of the simulation time.

Manager⁴ portion of the Autoresolver was enabled, scheduling aircraft for efficient arrival operations into over 200 of the busiest airports.

IV. Results

The four different coordination scenarios were run for both datasets resulting in 8 different ACES simulations. For the simulations where the enroute airspace was divided into different regions, approximately 20% of aircraft conflicts occurred when the two aircraft were being controlled by different Autoresolver agents. These conflicts required inter-Autoresolver coordination and the remaining 80% were handled by an individual Autoresolver instance. For reference, using a 12 core machine, the simulations where a single agent controlled all aircraft in the NAS took approximately three times as long as the corresponding parallelized solution.

Fig. 9 shows the total number of losses of separation in the simulation for the four different coordination cases and the two different flight data sets. It is important to note that, unlike previous studies,⁷ there are no filters applied to this loss-of-separation number. It is simply created by comparing the enroute portions of all trajectories and includes losses of separation at all enroute altitudes. The cases without coordination have a large number of losses of separation because the non-Autoresolver agent does not have information about aircraft in the Autoresolver-controlled airspace and vice versa. Including this external boundary coordination reduces the number of losses drastically to around 10 losses. Including intra-Autoresolver coordination in the simulation causes all these remaining losses to be resolved, with the same result when there is only one Autoresolver instance for the entire airspace.

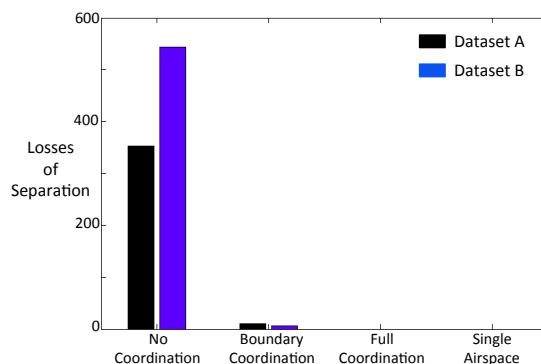


Figure 9. The total number of losses of separation for the four different cases.

The total numbers of resolutions required for the Autoresolver to maintain separation in its airspace for the different scenarios and datasets are shown in Fig. 10(a). Even though the no-coordination case has the most losses, it also has the most resolutions performed. This is because many short-range conflicts occur around the airspace border when the coordination is not present, and the Autoresolver must attempt to

solve those conflicts. The numbers of resolutions required in the full coordination case and the external boundary coordination case are approximately the same. A measure of the cost to dividing up the airspace and using the inter-Autoresolver coordination scheme is shown by comparing the full-coordination case and the single Autoresolver instance case for Dataset A. The single Autoresolver instance case has about 3.5% fewer resolutions required than the parallelized solution, probably related to the restricted visibility across airspace boundaries.

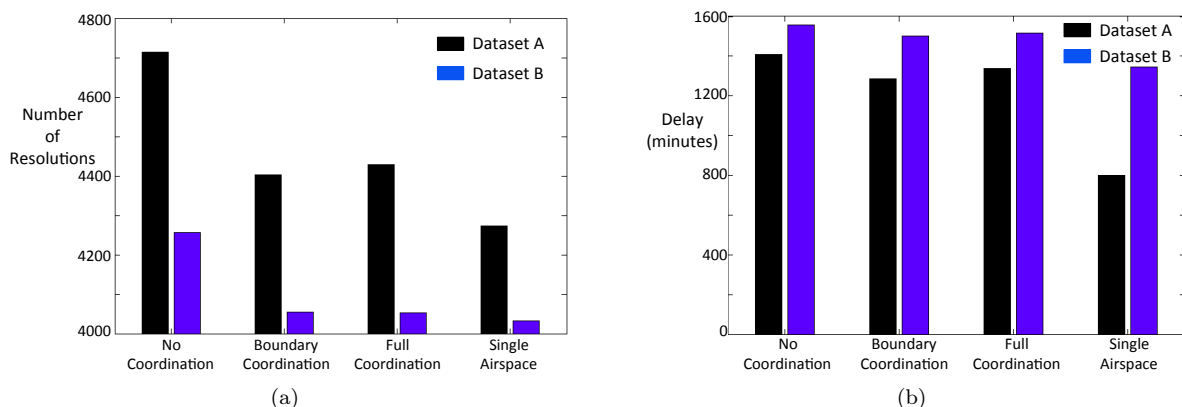


Figure 10. (a) The total number of resolutions performed by Autoresolver to maintain separation, and (b) the total system delay due to resolution maneuvers.

The most apparent trend when comparing the total resolution-related delay results shown in Fig. 10(b) with the total number of resolutions in Fig. 10(a) is that, while Dataset B required fewer resolutions in all cases than Dataset A, the opposite is true in terms of total delay. This is because Dataset B contained 28% arrival conflicts while Dataset A only contained 24% arrival conflicts, and arrival conflicts require a larger delay to solve. In fact, for the fully coordinated cases, the average delays per resolution for non-arrival conflicts were within one second of each other between the two datasets.

Another less apparent trend is that full coordination increased the total delay in both cases as compared to the boundary coordination cases, while the total number of resolutions remained approximately the same. This increase in the average delay per resolution for both demand sets is a direct consequence of the coordination where, for a subset of all conflicts, efficient resolutions to resolve the conflict are only created for one of the aircraft. This reduces the search space for efficient solutions and has a moderate impact on the overall system delay. To isolate the impact of the coordination more directly, the full coordination case (with divided airspace) and the single airspace case in Fig. 10(b) can be compared. As expected, using a single agent to manage the airspace resulted in lower delays, especially for Dataset A. Part of this total delay reduction is due to fewer total resolutions, but part of this is that the resolution process is more efficient on average. Fig. 11 shows histograms of delay distributions for the resolutions comparing the single airspace solutions (white bars) to the distributed solutions (solid bars). For both datasets, it can be seen that the distributed airspace solution is skewed further to the right (higher delay) than the single airspace solution indicating that on average, the coordination resulted in less efficient resolutions. An active coordination system, where agents negotiate the most efficient resolutions, could remove this added burden, but it would also add more complexity to the solution.

Finally, Fig. 12 shows the number of cases where two aircraft are maneuvered at the same time to resolve the same conflict scenario. Since the Autoresolver is designed to maneuver only one aircraft to resolve a conflict, this is an indication of the inter-Autoresolver coordination necessary. As shown in the figure, this is a relatively small percentage of the total resolutions required for the two non-Autoresolver coordinated cases, but all of the loss of separation cases in Fig. 9 were a result of each agent in a conflict assuming that the other agent would resolve the conflict repeatedly until the time of loss of separation. As expected, when Autoresolver coordination was added, there were no cases where two aircraft were maneuvered to resolve the same conflict.

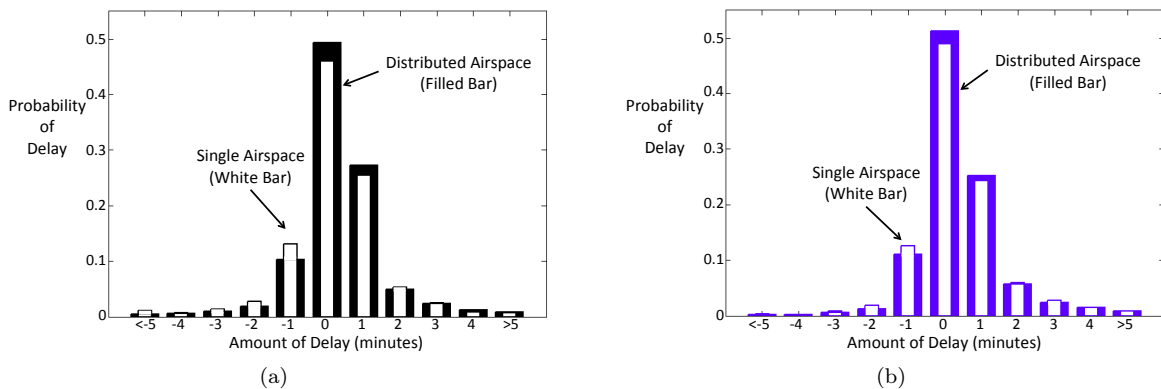


Figure 11. Histograms showing the distributions of delays for resolutions for the fully coordinated case (filled bars) and the single airspace case (white bars) for (a) Dataset A and (b) Dataset B.

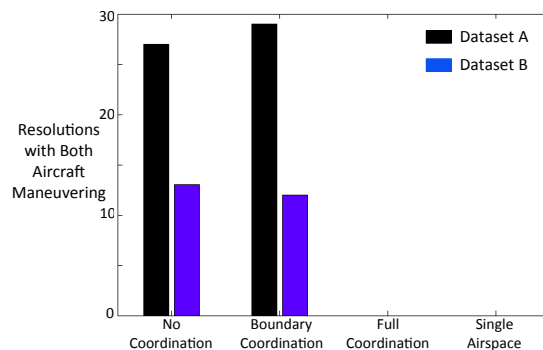


Figure 12. The total number of resolutions where two aircraft are maneuvered simultaneously to resolve the same conflict.

V. Conclusions

A future, automated separation assurance system may solve conflicts across the entire airspace in centralized fashion by one monolithic agent or, as is the case now, in a distributed fashion by agents located throughout the NAS. For redundancy and computational efficiency, the full problem will probably be decomposed, allowing for parallel solution. One decomposition of the problem was presented in this paper with one agent in each Center responsible for separation in its own geographic region. This decomposition requires coordination amongst the independent agents across the airspace borders. The inter-agent coordination scheme, tailored for the separation assurance problem presented in this paper was designed to minimize explicit coordination between the agents to the extent possible.

Since the NAS will always share boundaries with other international airspace operators and it is very probable that different types of airspace will be handled by different control algorithms. Another important coordination paradigm is between internal control agents and external ones. One such paradigm is presented in this paper where trajectories are shared between external agents and Autoresolver agents and checked against the current, internal trajectories to determine if a hand off is possible. Once a hand off is accepted the trajectory is considered frozen by the internal agent.

These coordination schemes were simulated using two different moderate demand sets to study how the coordination schemes impact the efficiency and safety of the system. Full coordination resulted in no losses of separation in the simulations, completely removing the small number of losses present when there was no coordination between internal agents. The reduction of the search space for a subset of the conflicts due to the coordination rules did result in slightly less efficient solutions overall. This inefficiency may be removed by including an active negotiation system between agents, but our results suggest that the added complexity and synchronization required may outweigh the efficiency improvements.

References

- ¹Nolan, M. S., *Fundamentals of Air Traffic Control*, Brooks/Cole Wadsworth, Pacific Grove, CA, 3rd ed., 1999.
- ²Pingali, K., Nguyen, D., Kulkarni, M., Burtscher, M., Hassaan, M. A., Kaleem, R., Lee, T.-H., Lenharth, A., Manevich, R., Méndez-Lojo, M., et al., “The tao of parallelism in algorithms,” *ACM SIGPLAN Notices*, Vol. 46, ACM, 2011, pp. 12–25.
- ³Erzberger, H., “Automated Conflict Resolution for Air Traffic Control,” *25th International Congress of the Aeronautical Sciences*, 2006.
- ⁴Erzberger, H., Lauderdale, T. A., and Cheng, Y., “Automated Conflict Resolution, Arrival Management and Weather Avoidance for ATM,” *27th International Congress of the Aeronautical Sciences*, Nice, France, 2010.
- ⁵Farley, T. C. and Erzberger, H., “Fast-Time Simulation Evaluation of a Conflict Resolution Algorithm Under High Air Traffic Demand,” *7th USA/Europe ATM 2007 R&D Seminar*, 2007.
- ⁶Meyn, L. A., “Nationwide Evaluation of a Conflict Resolution Algorithm,” *AIAA Aviation Technology, Integration, and Operations Conference*, No. AIAA 2009-6905, Hilton Head, South Carolina, 2009.
- ⁷Lauderdale, T. A., “The Effects of Passive Coordination on Distributed Separation Assurance,” *AIAA Aviation Technology, Integration, and Operations Conference*, No. AIAA 2009-6908, Hilton Head, South Carolina, 2009.
- ⁸Narkawicz, A., Munoz, C. A., and Maddalon, J. M., “A Mathematical Analysis of Air Traffic Priority Rules,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Indianapolis, Indiana, 2012.
- ⁹George, S., Satapathy, G., Manikonda, V., Palopo, K., Meyn, L., Lauderdale, T. A., Downs, M., Refai, M., and Dupee, R., “Build 8 of the Airspace Concept Evaluation System,” *AIAA Modeling and Simulation Technologies Conference*, No. AIAA-2011-6373, Portland, Oregon, 2011.