AIAA Guidance, Navigation, and Control Conference
08 - 11 August 2011, Portland, Oregon

AIAA 2011-6218

A Rescheduling Method for Conflict-free Continuous Descent Approach

Yi Cao*

Purdue University, West Lafayette, IN 47906-2045

Sivakumar Rathinam †

Texas A & M University, College Station, TX 77843-3123

Dengfeng Sun‡

Purdue University, West Lafayette, IN 47906-2045

This paper presents a rescheduling method for solving the conflicts between arriving aircraft which fly their preferred descent paths using the continuous descent approach. The proposed approach is a centralized control solution and based on the 4-D trajectory concept. Each aircraft calculates its 4-D trajectory and the Estimated Time of Arrival before initiating descent procedure, and reports the 4-D trajectory to the terminal area controllers at the destination airport. The reported 4-D trajectories are used for arrival sequencing. The objective is to minimize the total delay while keeping the inbound traffic conflict-free. The rescheduling problem is formulated as a Mixed Integer Linear Program and solved with CPLEX. The solution is the delay assignments which aircraft use to adjust the arrival time for conflict avoidance. A minimum in-trail separation is imposed between successive arrivals to meet the arrival rate of the airport. The proposed approach leads to a conflict-free inbound traffic. Most importantly, it benefits the continuous descent approach in a sense that the conflicts are solved without changing the user-preferred trajectories, and the CDA trajectories are keep intact. Simulation results reveal that the re-scheduling algorithm is able to de-conflict all the arriving aircraft with different prescribed horizontal separation settings in a full day terminal airspace operation at Newark Liberty International Airport. The runtime of the algorithm is within a reasonable time horizon. Benefits analysis indicates that 80 tonnes fuel and 638 minutes flight time are saved as a consequence of employing the conflict-free CDA.

*Ph.D Student, School of Aeronautics and Astronautics, Purdue University. Student member, AIAA. `cao20@purdue.edu`.

†Assistant Professor, Mechanical Engineering, Texas A&M University, `srathinam@tamu.edu`

‡Assistant Professor, School of Aeronautics and Astronautics, Purdue University. Member, AIAA. `dsun@purdue.edu`.

Nomenclature

A_i	=	aircraft i
i, j	=	index of aircraft
k	=	decision variable index
h	=	altitude
L	=	the maximum number of unite time allowed for delay assignment
N	=	the total number of aircraft landing at the airport in the planning time horizon
n, m	=	index of estimated time of arrival at the airport for aircraft i and j respectively
$P_i(t_0^i, t_n^i)$	=	4-D trajectory vector of aircraft i in time window $[t_0^i, t_n^i]$
$p_i(t^i, \varphi^i, \lambda^i, h^i)$	=	geographical coordinate of aircraft i at time t
ΔT	=	time interval
t	=	time step
w_k^i	=	the k^{th} binary decision variable for aircraft i
φ	=	latitude
λ	=	longitude

I. Introduction

As the advocation for green air traffic increases steadily, Continuous Descent Approach (CDA) draws more and more attention. Projects, like PARTNER, Sourdine, and OPTIMAL,[1–3] are particularly initiated for evaluating this environmentally friendly approach. It is widely accepted that CDA is hindered from practicing on a regular basis due to its negative impact on the airport throughput. In Ref. 4, it is reported that the landing interval increases from nominal 1.8 minutes to 4 minutes when employing CDA. Meanwhile, poor predictability of the trajectories of the aircraft also increases the risk of losing separation between arriving aircraft. Although there are many conflict resolutions, few of which are suitable for de-conflicting aircraft performing the CDAs, since the CDA is easily aborted due to tactical maneuvers, such as changes in heading and (particularly) vertical speed.

There is a trend in current air traffic control system in which flight crews are endorsed more responsibilities to plan their preferred trajectories. This development relies on advanced aviation concepts, like the onboard Flight Manage System (FMS), as well as the communication system which enables reliable datalink between aircraft and control center. Ren, et al. analyzed the utilization of Lateral Navigation (LNAV) and Vertical Navigation (VNAV) functions of the FMS in CDA aircraft separation problem,[5] both in simulations and in a field test in September 2004 at Louisville International Airport (PARTNER project, U.S.). By defining optimal VNAV paths, flight tests verified that CDA can be successfully flown with safe separation. Later, a FMS prototype with 4-Dimensional (4-D) capabilities was developed, known as the ARINC 429 Bus, and equipped on the B737 and tested in the CASSI Flight Trials at Stockholm Arlanda Airport in 2009 (EUROCONTROL). The 4-D FMS was able to predict the 4-D trajectory, report current aircraft state information with a 2 Hz frequency, and navigate the aircraft through waypoints under required time of arrival (RTA) constraints with a precision of 6 seconds. These initiatives indicate the substantial progress that has been made in the recent years towards realizing the goal of trajectory based operations. The predictability of CDA is expected to improve as the technologies based on the trajectory based operations become more realizable. As a result, conflicts that arise due to CDA can be strategically solved without resorting to aborting the CDA in mid-flight.

The previous work related to the CDA in the literature have mainly focussed on the evaluation of the benefits of the CDA on the noise abatement, fuel savings, emission reduction etc. We are not aware of any contributions that strategically resolve conflicts between arrival aircraft flying CDAs. The objective of this article is to present a fast, re-scheduling algorithm for this strategic conflict resolution problem involving aircraft flying CDAs. Simulation results with flight data are presented to examine the feasibility of the proposed algorithm, and the potential benefits that could be obtained by implementing it. Specifically, the algorithm was implemented with a full day's flight data collected at the Newark Liberty International Airport. The results indicate that 80 tonnes of fuel and 638 minutes of flight time can be potentially saved as a consequence of implementing the conflict-free CDA.

This article is organized as follows: Section II introduces the concepts of CDA and 4-D trajectory. Section III discusses the strategic conflict resolution problem and the re-scheduling algorithm. Section IV presents the simulation results which validate the re-scheduling algorithm along with potential benefits. The article concludes with future work in Section V.

II. Continuous Descent Approach and 4-D Trajectory Based Operations

An ideal vertical profile of CDA is simulated using the Future ATM Concept Evaluation Tool (FACET)[7] and shown in Fig. 1. The CDA keeps the aircraft cruising at a relatively high altitude as long as possible. When arriving at the top of descent (TOD), the aircraft descends along a smooth slope with engines running at idle or near-idle settings which require significantly less engine thrust. Compared to the conventional step-down descent, the CDA reduces[8] fuel consumption, emission, and noise impact by avoiding low altitude (between 3,000 ft and 10,000 ft) level flight before intercepting the final 3 degree glidepath (normally 8 to 25 nautical miles away from the touchdown depending on local circumstances).[9] It also saves flight time due to the longer high altitude cruise with high speed. However, regardless of the environmental benefits, to date the CDA has yet been a nationwide practice due to airport safety and efficiency concerns. The near idle throttle settings makes the descent trajectories less predictable. As a result, air traffic controllers (ATCs) have to block large chunk of airspace to separate the landing aircraft. This leads to a reduction in the throughput of an airport. So far, the CDA has been practiced only in low-density conditions or applied to a selected portion of inbound traffic in the nighttime hours rather than on a standard terminal airspace operation basis.[8,10–12]

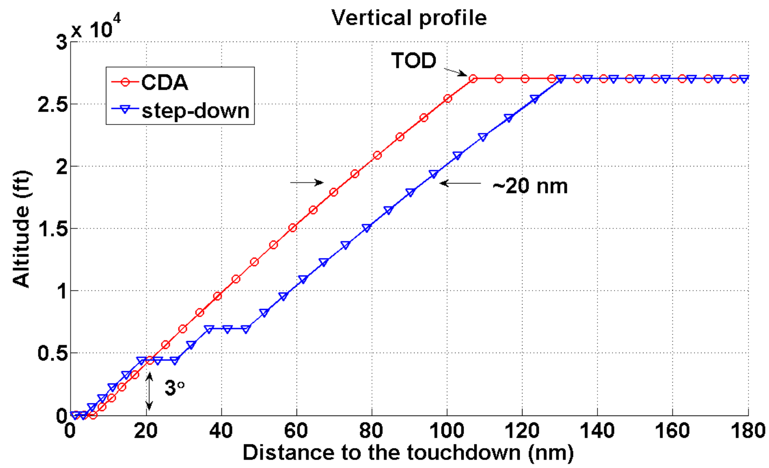


Figure 1. CDA and step-down profile simulated by FACET

The 4-D Trajectory Based Operations is a concept that can enhance the predictability of the CDA.[13] Essentially, in this concept, each aircraft has a capability to accurately follow a planned, feasible trajectory that has been agreed upon with the controllers. The operations are then based on these trajectories that specify the complete set of positions of each aircraft along with its time of arrival at each position. The realization of this concept strongly depends on the on-board FMS and a reliable datalink between the aircraft and ATCs.[14] Aircraft equipped with the P-RNAV system is able to fly a 4-D trajectory accurately be within +/-1 NM for 95% of the flight time.[15] En route flights calculate its estimated time of arrival (ETA) and planned standard terminal arrival routes (STARs), then report to the ATCs for approval via dedicated datalink. ATCs send back the RTAs and STARs which help the FMS to fine tune the accurate ETAs. Such negotiations are initiated at least one hour before reaching the TOD, allowing sufficient time for the ATCs to sequence and merge the arriving flows.

III. The Strategic Conflict Resolution Model and Algorithms

Any tactical maneuver for de-confliction purposes may require extra thrust inputs to the aircraft, and therefore, should be avoided. This article targets at solving the conflicts between arriving flights without disturbing the CDA procedures. 4-D trajectory is a requirement facilitating strategic planning. Throughout this article, it is assumed that flights under control are able to calculate and report their 4-D trajectories, which forms the basis for the proposed re-scheduling algorithms.

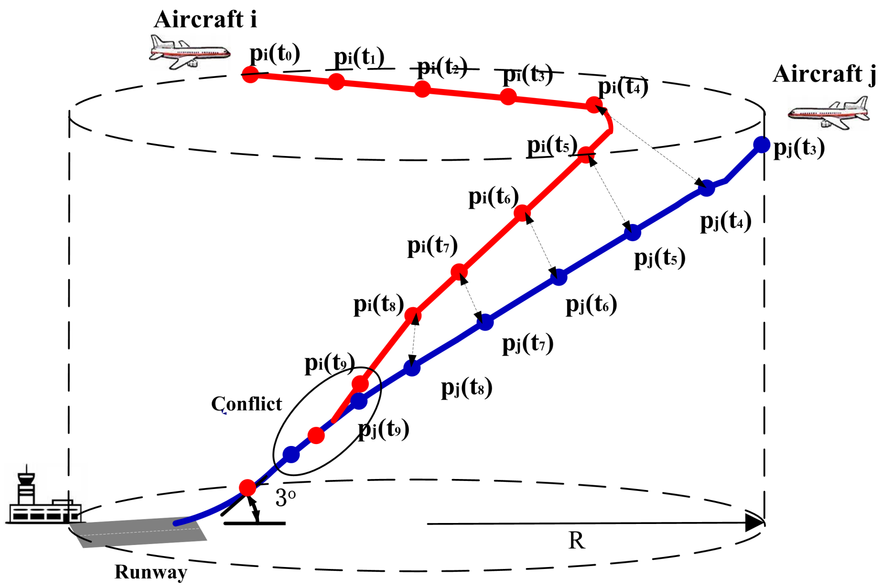


Figure 2. Illustration of 4-D trajectory

A cylindrical-shaped region with a radius R centered around the airport is defined, as shown in Fig. 2 (note that Fig. 2 is not drawn to scale, the airport should be at the center of the cylinder). Airspace within this cylinder is of interest and hereafter referred as the terminal area. R should be determined in such a way that majority of the flights begin the CDAs within the cylinder. As a result, their descent trajectories are covered by the model considered in this article. Current runway usage follows a first-come-first-served principle. All flights are required to report their ETAs to the ATCs at the destination airport. If the arriving aircraft are allowed to freely plan their own landing without considering mutual inferences, it is quite likely that two or more aircraft will pass the same airspace within a small time window. In fact, a simulation based on FACET reveals that uncontrolled flight planning leads to several conflicts, majority of which occur at low altitude ($< 3,000\ ft$), as shown in Fig. 3. It is necessary for the ATCs to issue the RTAs in order to stagger the arriving flights such that successive arrivals are safely separated, and arrival rate does not exceed the capacity of the airport. Rescheduling the ETAs does not change the planned positions of the aircraft but simply imposes delay controls on flights exposed to potential conflicts. Impacted flights postpone their entries into the terminal area until the airspace they require is cleared, and then follow their descent trajectories as planned. This type of delay control essentially removes all the conflicts and preserves the characteristics of CDA. The following subsections outline the details involved in the conflict resolution model.

A. Conflict Resolution Model

Let the set of aircraft be denoted as $[A_1, A_2, \cdots, A_N]$. The 4-D trajectory of aircraft of A_i is denoted by:

$$P_i(t_0^i, t_n^i) = [p_i(t_0^i, \varphi_0^i, \lambda_0^i, h_0^i), p_i(t_1^i, \varphi_1^i, \lambda_1^i, h_1^i), \cdots, p_i(t_n^i, \varphi_n^i, \lambda_n^i, h_n^i)], \tag{1}$$

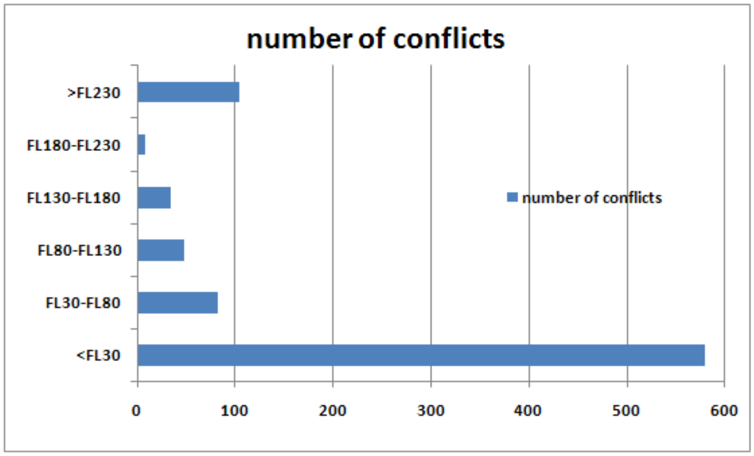


Figure 3. Conflicts distribution in the terminal area without control strategies. A conflict of two aircraft lasting for one minute is counted for one conflict.

where t_0^i is the estimated time of arrival for A_i at the boundary of the cylinder, and t_n^i is the estimated time of landing for A_i at the airport. $P_i(t_0^i, t_n^i)$ essentially consists of all the positions of aircraft A_i at times that are separated by a sampling interval ΔT. Similarly, say the 4-D trajectory of A_j is also defined as follows:

$$P_j(t_0^j, t_m^j) = [p_j(t_0^j, \varphi_0^j, \lambda_0^j, h_0^j), p_j(t_1^j, \varphi_1^j, \lambda_1^j, h_1^j), \cdots, p_j(t_m^j, \varphi_m^j, \lambda_m^j, h_m^j)], \tag{2}$$

$[t_0^i, t_n^i]$ and $[t_0^j, t_m^j]$ are the time windows when A_i and A_j are present in the terminal area. If $[t_0^i, t_n^i]$ and $[t_0^j, t_m^j]$ intersect in some period, e.g. $[t_0^j, t_m^j] \cap [t_0^i, t_n^i] = [t_p, t_q]$, flight legs $P_i(t_p, t_q)$ and $P_j(t_p, t_q)$ must be checked for potential conflicts. A conflict detection function is defined as follows:

$$C(p_i(t, \varphi_t^i, \lambda_t^i, h_t^i), p_j(t, \varphi_t^j, \lambda_t^j, h_t^j,)) = \begin{cases} 0 & if\ |h_t^i - h_t^j| \geq H \\ 1 & if\ (|h_t^i - h_t^j| < H)\ \&\&\ (d < D) \end{cases} \tag{3}$$

where H and D are the minimum vertical separation and horizontal separation respectively. Usually, $H =$ 1000 ft if aircraft are below FL290, and $H = 2000$ ft if aircraft are above FL290. $D = 5$ nm (nautical mile). d is the great-circle distance between A_i and A_j, calculated using the *Haversine formula* (it is assumed a spherical earth and ignore the ellipsoidal effects):

$$d = 2r \times arcsin\left(\sqrt{sin^2(\frac{\Delta\varphi}{2}) + cos\varphi_i cos\varphi_j sin^2(\frac{\Delta\lambda}{2})}\right)$$

where r is the radius of the earth, usually $r = 3440.07\ nm$. Now, an *aggregate*, conflict detection function is defined as follows:

$$CD(i, j, t_p, t_q) = \sum_{t=t_p}^{t_q} C(p_i(t, \varphi_t^i, \lambda_t^i, h_t^i), p_j(t, \varphi_t^j, \lambda_t^j, h_t^j)) \tag{4}$$

If there are conflicts in time window $[t_p, t_q]$, it follows that $CD(i, j, t_p, t_q) > 0$. In this case, the initial estimated arrival time t_0^i or t_0^j must be rescheduled to ensure that A_i and A_j are safely separated. For example, suppose A_i is delayed by Δt, then a rescheduled 4-D trajectory of A_i is generated as follows:

$$P_i(t_0^i + \Delta t, t_n^i + \Delta t) = [p_i(t_0^i + \Delta t, \varphi_0^i, \lambda_0^i, h_0^i), p_i(t_1^i + \Delta t, \varphi_1^i, \lambda_1^i, h_1^i), \cdots, p_i(t_n^i + \Delta t, \varphi_n^i, \lambda_n^i, h_n^i)] \quad (5)$$

Note that $P_i(t_0^i + \Delta t, t_n^i + \Delta t)$ is simply a shift of the original $P_i(t_0^i, t_n^i)$ in time, *i.e.*, A_i will pass the original waypoints with a delay of Δt. Due to the delay, the intersection of time window is changed to $[t'_p, t'_q]$. If the trajectory of the aircraft is sufficiently shifted, then, one can ensure that $[t'_p, t'_q] = \emptyset$, or $CD(i, j, t'_p, t'_q) = 0$. Essentially, one can resolve the conflict between the aircraft by suitably delaying the entry time of the aircraft into the boundary. It is also possible to resolve a conflict by requiring the aircraft to arrive at the boundary earlier than the ETAs. However, we do not adopt this approach as it may not always be feasible to increase the speed of the aircraft as there may be airspeed constraints. As it is always feasible to hold an aircraft in the air (circling at most) to meet a late RTA, delaying the entry time of an aircraft is the only control action considered in this study.

B. Mixed Integer Linear Program formulation

Resolving conflicts when there are multiple aircraft involved is generally complicated as resolving a conflict between two aircraft may cause additional conflicts with other aircraft. Therefore, a global solution is necessary. The strategic conflict resolution problem is to find a global delay assignment for each aircraft such that no conflict will happen in the terminal area. Inspired by the Bertsimas and Stock-Patterson model,[16] this problem is formulated as a Mixed Integer Linear Program (MILP) as discussed below.

w_k^i is the binary decision variable that is used to denote the amount of delay that must be assigned to aircraft A_i. $w_k^i = 1$ if aircraft A_i is delayed by $k\Delta T$ and is equal to 0 otherwise. The objective of the conflict resolution problem is to find the delay that must be assigned to each aircraft so that an objective that depends on the weighted sum of all the delays is minimized. The objective function we consider is denoted by:

$$f = \min \sum_{i=1}^{N} \sum_{k=0}^{L} c^i w_k^i k \Delta T \quad (6)$$

where c^i is a weight imposed on a particular flight, which is useful when considering the fairness and efficiency among flights. For example, if delaying a B777 is equivalent to delaying a Learjet45 to solve a conflict, it is desirable to delay the latter for less fuel consumption. If delaying a commercial flight carrying passengers is equivalent to delaying a cargo flight, it is desirable to delay the latter for minimizing the passengers' delay.

The objective function is subject to the following constraints:

1. One delay decision for each aircraft:

$$\sum_{k=0}^{L} w_k^i = 1, \qquad i \in [1, N] \quad (7)$$

2. Binary decision variable:

$$w_k^i \in [0,1], \qquad i \in [1,N], k \in [0,L] \tag{8}$$

3. Conflict detection:

$$\begin{aligned}
[t_p, t_q] = [t_0^i + w_{k_i}^i k_i \Delta T, t_n^i + w_{k_i}^i k_i \Delta T] \quad &\cap \quad [t_0^j + w_{k_j}^j k_j \Delta T, t_m^j + w_{k_j}^j k_j \Delta T]) \\
\textbf{if } ([t_p, t_q] \neq \emptyset \quad &\&\& \quad CD(i,j,t_p,t_q) > 0) \\
\textbf{then } \text{set: } w_{k_i}^i + w_{k_j}^j \quad &\leq \quad 1 \\
i,j \in [1,N], \qquad & \qquad k_i, k_j \in [0,L]
\end{aligned} \tag{9}$$

Ineq. (9) can be explained as follows: Suppose, let both the aircraft A_i and A_j be assigned delays such that $w_{k_i}^i = 1$ and $w_{k_j}^j = 1$ respectively. If such an assignment potentially incurs a conflict, then the above constraint ensures it is not feasible.

In the context of the scheduling literature, the above conflict resolution problem can be viewed as a generalization of a single-machine scheduling problem[19] with sequence-dependant processing times for the jobs.

C. Re-scheduling Algorithm

Table 2 summarizes the re-scheduling algorithm. Initially, a simulation is run to generate the planned 4-D trajectories, which emulates the ETA calculation in real-world operations. Since each aircraft plans its own trajectory, they compete for usage of airspace and runway. Essentially, the algorithm reorders the times of arrival of the inbound traffic and ensures that the traffic is conflict-free. It is worth noting that the maximum delay L is crucial to the existence of a feasible solution. If L is too small, it is not able to separate a set of conflicted aircraft even with the maximum delay. If L is too large, there must be a feasible solution (in the worst case, flights are sequenced to fly into the terminal area one by one). However, the dimension of the problem could grow to an extent that it may be computationally very difficult to solve. Hence, to search for a minimum feasible delay assignment, we start the simulation by choosing a small value of L (≈ 10). If the optimization is infeasible, the algorithm increases L by one, and starts a new run. The algorithm does not stop until there is a feasible solution.

D. Minimum in-trail separation

The re-scheduling algorithm solves conflicts in the descent phase, but does not consider another important constraint, namely the arrival rate of the aircraft at an airport. Each airport has a maximum bound on the landing rate of aircraft depending on the runway configurations and local terrain. Therefore, the arrival flow must be sequenced so that the landing rate of the aircraft satisfies this bound. This requirement on the landing rate can be implemented by requiring a minimum in-trail separation between successive arrivals. For example, the typical benchmark rate of EWR is 40 flights per hour,[18] which is equivalent to a 1.5 minute separation between successive flights. Essentially, the solution to the conflict resolution problem must be modified so that the landing intervals are not less than this minimum in-trail separation.

From the notations, it follows that the landing time of aircraft A_i is:

$$t^i_{landing} = t^i_n + \sum_{k=0}^{L} w^i_k k \Delta T$$

First, the time of arrivals are sorted in a non-decreasing order in terms of $t^i_{landing}$ using the *Bubble sort* algorithm. Then successive arrivals are checked for the minimum in-trail separation. Suppose that A^j lands next to A^i:

$$\textbf{if } (t^j_{landing} - t^i_{landing} < 1.5)$$
$$t^j_{landing} = t^i_{landing} + 1.5$$

This process finally produces a arrival sequence that respects both the conflict separation constraints and arrival rate constraints.

IV. Model Validation

This section presents simulations validating the proposed algorithm based on FACET. The re-scheduling algorithm is coded in C++ and run on a 2.8 GHz INTEL i7 CPU, 16G RAM DELL workstation running LINUX. The optimization solver used is CPLEX11.0,[20] which is capable of solving an Integer Program. Historical nonmilitary flight data are acquired from ASDI/ETMS,[17] which includes latitude, longitude, altitude and time information of all airborne aircraft in the United States. Newark Liberty International Airport (EWR) is selected as it is one of the busiest airports in the U.S.. A full day (March 15, 2005) inbound traffic at EWR is simulated with $\Delta T = 1$ *minute*. There were 721 aircraft landing at EWR on that day.

A. Simulation setup

The validation consists of two simulations. In the first simulation, aircraft perform unconstrained CDAs without rescheduling. Each aircraft is navigated by the filed flight plans, and descends along the course of CDA adhering to the built-in performance database. The first simulation generates the 4-D trajectories of all flights. The conflict count obtained from this case serves as a baseline. In the second simulation, aircraft perform constrained CDAs under separation constraints and airport arrival rate constraint. R is set to be 180 *nm* such that majority of the TODs lie inside the yellow circle in Fig. 4. As a result, majority of the CDAs are not interrupted as delay controls are finished outside the terminal area. The 4-D trajectories exported from the first simulation are fed into the re-scheduling algorithm. The MILP is solved by CPLEX. Once the optimization is done, the optimal delay solutions are used to run the second simulation. In both simulations, statistics of fuel consumption and flight time are recorded, as well as the conflict information.

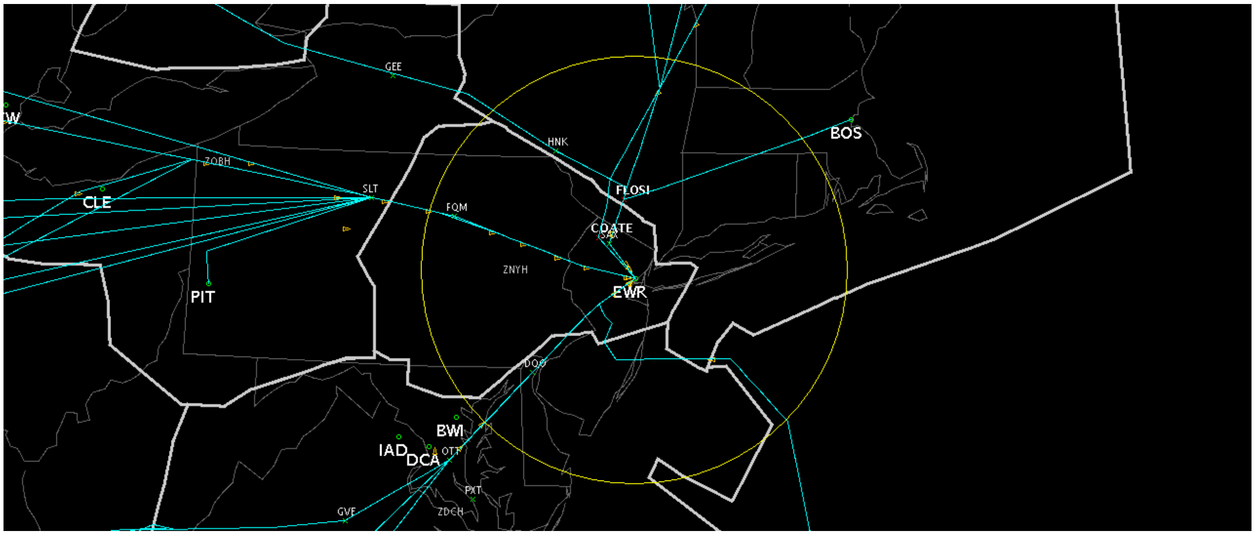
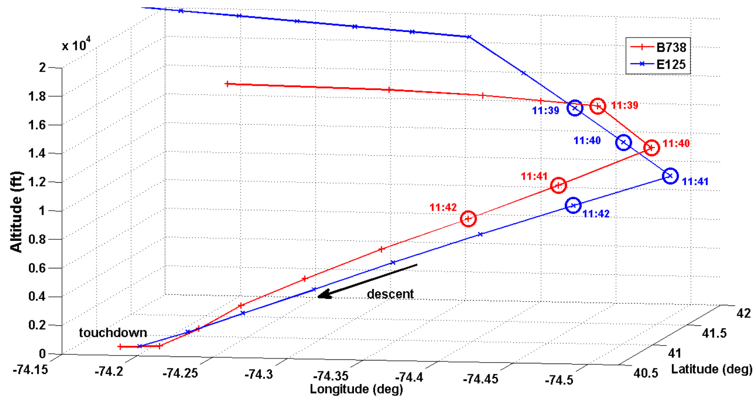


Figure 4. A snapshot of terminal area at the Newark Liberty International Airport (EWR). The yellow circle is the boundary of the terminal area. Blue lines are some filed flight plans of the arriving flights.

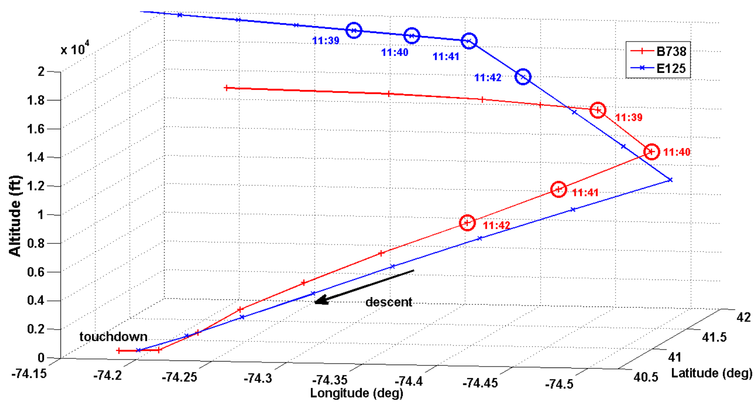
B. Conflict resolution

FACET provides a built-in conflict detection module to count the conflict using a prescribed separation criterion. The conflict is counted in such a way that, for example, the conflict count increases by 5 if loss of separation between two aircraft occurs and lasts for 5 minutes. Simulation results reveal that there are 736 conflicts reported in the unconstrained CDA. In constrained CDA, all conflicts are resolved, indicating that the re-scheduling algorithm is effective in solving the conflict. Figure 5 demonstrates one of the cases in which two aircraft are staggered by delay control. Figure 5(a) shows the 4-D trajectories of two aircraft in the unconstrained CDA. An E125 is in conflict with a B738 starting at 11:39 AM and ending at 11:42 AM. Positions where conflict occurs are highlighted by circles and time stamps. Figure 5(b) shows the rescheduled 4-D trajectories. The re-scheduling algorithm issues a delay of 4 minutes to E125, while B738 still adheres to its original time of arrival. It can be seen that these two aircraft are staggered in time. Original conflict is solved as a result. Moreover, recorded conflict information reveals that E125 is also in conflict with other aircraft which is not shown Fig. 5. A 4 minutes delay guarantees that all conflicts associated with E125 are solved.

Figure 6 shows the arrival rates in unconstrained CDA and constrained CDA. Compared to unconstrained CDA, the arrival rate in constrained CDA has little change before 15:00 PM when the traffic is at low level. A high level landing demand comes after 15:00 PM. It is observed that the arrival rate exceeds the airport benchmark rate during the peak hours in unconstrained CDA. In constrained CDA, the minimum in-trail separation restricts the arrivals. Many aircraft are sequenced to land such that the arrival rate keeps at its maximum bound. Previous simulations use the enroute horizontal separation of 5 nm. By FAA rules,[21] 3 nm is common in terminal airspace at low traffic level. But this study evaluates CDA in high density



(a)



(b)

Figure 5. Two aircraft in is separated by rescheduling

traffic environment. Insufficient separation may lead to accidents due to wake turbulence. On the other hand uncertainties are common in real traffic and generally are not taken into account a-priori. Therefore, a larger separation is expected to provide a buffer to accommodate the uncertainties. Table 2 shows the statistics varying against the increased horizontal separation. As the value of H increases, the runtime (includes problem generation, optimization, and sequencing) almost exponentially increases. This is due to the increased dimension of decision variables. Larger separation requires longer delay to space the aircraft. The longer an aircraft is delayed, the more aircraft it is likely to interfere with. Such chain effects makes the number of constraints (particularly, Ineq. 9) increase rapidly, which in turn requires a bigger value of L. But in all cases we implemented, the algorithm yields a solution within a reasonable time horizon.

The total delay comprises delay for CDA separation and delay for in-trail separation. The latter accounts for approximately 90% of the total delay. It is worthy noting that the CDA delay doubles or even triples as the horizontal separation increases, but the delay for in-trail separation has relatively small changes. This suggests that this part of delay is not a consequence of employing CDA, and can be removed by reasonably adjusting the departure time of aircraft.

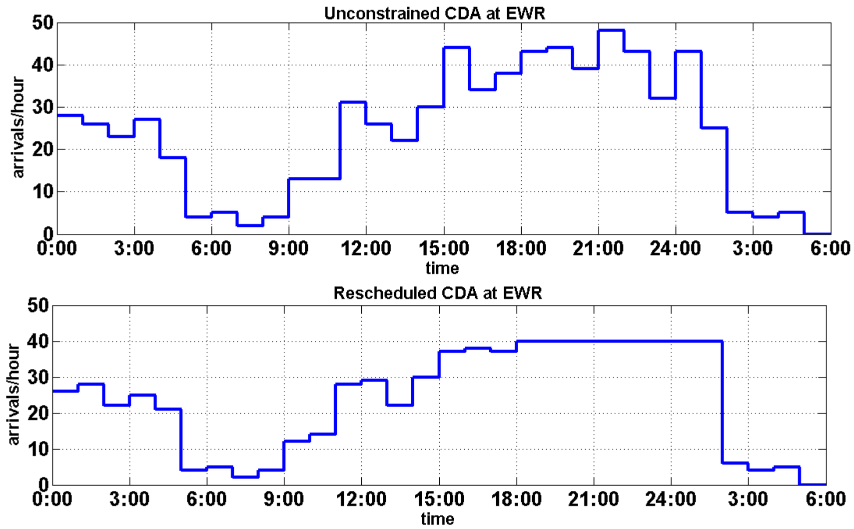


Figure 6. Arrival rate comparison (Horizontal separation = 5nm)

C. Discussion of benefits and trade-offs

Compared to the conventional step-down approach, CDA reduces the flight time as well as fuel consumption. Statistical analysis reveals that total 80,027 kg fuel (110 kg/flight) and 638 minutes flight time (53 sec/flight) are saved (with $D = 5$ nm) if the step-down descents are changed to CDAs in a full day operation. As a trade-off, a total delay of 431 minutes is required for CDA separation. Depending on the scenario, this delay can be absorbed by delaying the aircraft either on the ground or while it is airborne. For instance, if delaying an aircraft on the ground does not add any constraints at its departing airport, then the aircraft can be held on the ground. On the other hand, if the cost due to delaying the aircraft while it is airborne is manageable, then one consider the later option.

V. Conclusion

A re-scheduling algorithm is proposed to strategically solve the conflicts between the arriving flights flying the CDAs. This approach separates the aircraft by rescheduling their times of arrival at the boundary of the terminal area. A minimum in-trail separation is imposed to guarantee the capacity benchmark rate of the airport is not violated. Simulation results verify that, in a single airport scenario, all conflicts can be solved with a reasonable runtime, and the airport capacity is well respected. Compared to other conflict resolutions, this approach does not require tactical maneuvers which potentially disturb the CDA procedure. Although it results in delays, the savings of fuel and flight time at the terminal area render such delays a minor trade-off.

References

[1]Clarke, J. P. B., et al., "Development, Design, and Flight Test Evaluation of a Continuous Descent Approach Procedure for Nighttime Operation at Louisville International Airport," Report No. PARTNER-COE-2005-02, Jan. 9, 2006.

[2]Sourdine II final report, URL: `http://www.sourdine.org` [cited 12 Jan 2010].

[3]"OPTIMAL, D2.2-1 Aircraft procedures definition-ACDA," Document ID: WP2-NLR-022-V1.2-TW-CO. URL: `http://www.optimal.isdefe.es/public/publications/CDA.html`[cited 12 Jan 2010].

[4]Mohleji, S.D., and et al., Curved approaches in the Netherlands: feasibility and benefits, MITRE Technical report MTR99W99W122, 1999.

[5]Ren, L., and Clarke, J-P. B., "Flight Demonstration of the Separation Analysis Methodology for Continuous Descent Arrival" 7^{th} *USA/Europe ATM 2007 R&D Seminar*, Barcelona, Spain, 2-5 Jul. 2007 .

[6]"CTA/ATC System Integration Studies (CASSIS) Flight Trials Report," CASSIS project, D-LFV2008-, Rev 01.00, Mar. 27, 2009.

[7]Bilimoria, K. D., Sridhar, B., Chatterji, G. B., Sheth, K. S., and Grabbe, S. R.,"FACET: Future ATM Concepts Evaluation Tool," *Air Traffic Control Quarterly*, Vol. 9, No. 1, 2001, pp. 1-20.

[8]Clarke, J.-P. B., Ho N. T., Ren L., Brown, J., Elmer, K. R., Tong, K. O., Wat, J. K., "Continuous Descent Approach: Design and Flight Test for Louisville International Airport," *Journal of Aircraft*, Vol. 41, No. 5, Sep-Oct 2004.

[9]"Continuous Descent Approach Implementation Guidance Information," ICAO CDA Implementation Manual, May 2008.

[10]Wat, J., Follet, J., Mead, R., Brown, J., Kok, R., Dijkstra, F., Vermeij, J., "In Service Demonstration of Advanced Arrival Techniques at Schiphol Airport," 6*th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Wichita, Kansas, 25-27 Sep. 2006.

[11]Reynolds, T. G., Ren, L., Clarke, J. P. B., Burke, A. S., Green M., "History, Development and Analysis of Noise Abatement Arrival Procedures for UK Airports," *AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO)*, Arlington, Virginia, 26-28 Sep. 2005.

[12]"Preliminary Analysis of Continuous Descent Approaches into Ottawa Macdonald-Cartier International Airport," SRS-TSD-003 Rev. 0, Specific Range Solutions Ltd., Nov 27, 2009.

[13]Thomas Prevot, Vernol Battiste, Everett Palmer, Stephen Shelden, "Air Traffic Concept Utilizing 4D Trajectories and Airborne Separation Assistance," *AIAA Guidance Navigation and Control Conference*, Austin, Texas, Aug. 2003, AIAA-2003-5770-GNC.

[14]Coppenbarger, R. A., Mead, R. W., Sweet, D. N., "Field Evaluation of the Tailored Arrivals Concept for Datalink-Enabled Continuous Descent Approach," 7^{th} *AIAA Aviation Technology, Integration and Operations Conference* , 18-20 September 2007, Belfast, Northern Ireland.

[15]EUROCONTROL Navigation Domain - PRNAV Home. URL: `http://www.ecacnav.com/content.asp?CatID=201` [cited 12 Jan 2010].

[16]Bertsimas, D., and Patterson, S.S. "The Air Traffic Flow Management Problem with En route Capacities," *Operations Research*, Vol. 46, No. 3, May - Jun., 1998, pp. 406-422.

[17]"Enhanced Traffic Management System (ETMS)," Volpe, National Transportation Center, U.S. Department of Transportation, Cambridge, MA, Tech. Rep VNTSC-DTS56-TMS-002, October 2005.

[18]Airport Capacity Benchmark Report 2004, URL: `http://www.faa.gov/about/office_org/headquarters_offices/ato/publications/bench/DOWNLOAD/pdf/EWR_2004.pdf` [cited 12 Jan 2010].

[19]Brucker, P., *Scheduling Algorithms*, fifth edition , Springer, Heidelberg, 2007.

[20]ILOG CPLEX 11.0 Users Manual, URL: `http://www.decf.berkeley.edu/help/apps/ampl/cplex-doc/` [cited 12 Jan 2010].

[21]FAA Order 7110.65S, 5-5-4 Radar Separation Minima.

Table 1. Re-scheduling algorithm

1. Run simulation:

get planned 4-D trajectory of A_i:

$[t_0^i, t_n^i]$ and $P_i(t_0^i, t_n^i) = [p_i(t_0^i, \varphi_0^i, \lambda_0^i, h_0^i), p_i(t_1^i, \varphi_1^i, \lambda_1^i, h_1^i), \cdots, p_i(t_n^i, \varphi_n^i, \lambda_n^i, h_n^i)]$

2. Initialization:

$L =$ small positive integer

$N =$ number of aircraft

3. Formulation and Iteration:

do{

$L = L + 1$

for $i = 1 : N$

define decision variable: $w^i = [w_0^i, w_1^i, \ldots, w_L^i]$

for $k = 0 : L$

generate interval: $[t_0^i + w_k^i k\Delta T, t_n^i + w_k^i k\Delta T]$

generate delayed 4-D trajectory: $P_i(t_0^i + w_k^i k\Delta T, t_n^i + w_k^i k\Delta T)$

end for

end for

formulate: $f = \min \sum_{i=1}^{N} \sum_{k=0}^{L} c^i w_k^i k\Delta T$

$s.t.\ \sum_{k=0}^{L} w_k^i = 1$

$w_k^i \in [0, 1]$

$[t_p, t_q] = [t_0^i + w_{k_i}^i k_i \Delta T, t_n^i + w_{k_i}^i k_i \Delta T] \cap [t_0^j + w_{k_j}^j k_j \Delta T, t_m^j + w_{k_j}^j k_j \Delta T])$

if $([t_p, t_q] \neq \emptyset$ && $CD(i, j, t_p, t_q) > 0)$

$w_{k_i}^i + w_{k_j}^j \leq 1, \qquad i, j \in [1, N]$

end if

run optimization.

} ***while***(infeasible solution)

4. Output: $w^i = [w_0^i, w_1^i, \ldots, w_L^i], \qquad i \in [1, N]$

Table 2. Algorithm outputs with different horizontal separations

D (nm)	Runtime (sec)	L (min)	No. of Delayed AC	Max. Individual delay(min)	Total CDA delay (min)	In-trail separation delay (min)	Total delay (min)
5	1.98	11	589	34.5	431	6451	6882
6	6.35	13	591	365	600	6528	7128
7	14.76	15	597	36.5	749	6585.5	7334.5
8	67.83	17	603	37.5	962	6417.5	7377.5
9	221.92	18	606	45	1185	6243.5	7428.5