

DESIGN CONSIDERATIONS FOR SHORTCUT PATH-BASED TIME RECOVERY

Shannon J. Zelinski, NASA Ames Research Center, Moffett Field, California

Abstract

This paper introduces the concept of using tactical shortcuts to improve arrival precision and thereby increase throughput. This concept schedules flights to a longer path for which a shortcut path option is available for time recovery when speed is restricted. Shortcut design parameter sensitivity for schedule-based and spacing-based use policies is explored for single vs. multiple shortcut designs and shortcut availability. Simulation results show schedule-based shortcuts can reduce scheduling buffer size by 35-55% (increasing maximum throughput) or reduce separation violation probability 15-20% (reducing controller workload). Spacing-based shortcuts can reduce scheduling buffer size or separation violation probability by an additional 20% or 5% respectively. Additional studies were conducted to evaluate throughput and delay performance of each shortcut use policy for a wide range of demand rates. Spacing-based shortcuts performed the best. A use policy that combined the schedule-based shortcut method and scheduling without shortcuts outperformed using either of these methods individually. The potential benefits of using tactical shortcuts warrants future study in its application to multi-point scheduling, performance-based operations, geometric design, and decision support tools.

Introduction

With today's congested airports, a considerable amount of research is focused on increasing airport throughput. The throughput of a single arrival runway is constrained by how tightly flights can be spaced as they complete the final approach. Whereas wake separation requirements dictate a minimum allowable spacing for safety, the achieved spacing for managing capacity is nominally a half to one mile greater to account for imprecision associated with aircraft merging and wind uncertainties [1]. The current practice of vectoring to properly space merging flows is highly flexible, but demands a high degree of controller attention, and produces relatively

varied spacing results. As such, other concepts have been explored.

Time-based arrival management concepts such as DLR's 4D Cooperative ARrival MAnager (4D-CARMA) [2-4] and NASA's Terminal Arrival Precision Scheduling and Spacing (TAPSS) [5-7] make use of improved estimated-time-of-arrival (ETA) afforded by fixed routing, such as Area Navigation (RNAV) and Required Navigation Performance (RNP), and wind forecasts to develop precise arrival schedules. The question of spacing precision then becomes one of schedule conformance. The more precisely flights can be controlled to meet their scheduled times of arrival, the tighter flights may be scheduled. To this end, these time-based arrival management concepts employ controller decision support tools to help meet the schedule. Higher schedule conformance can be achieved by controlling aircraft to their scheduled time of arrival with speed control rather than vectoring [8]. The increased schedule conformance allows reduced scheduling buffers and increased throughput.

However, speed control alone offers a limited range of flexibility to respond to non-compliance to schedule, especially when time recovery is needed, due to maximum and minimum speed restrictions. The tactical use of shortcut path options is another method of time recovery for late arrivals that could increase schedule conformance, allowing the precision schedulers to reduce their buffers and increase throughput even further.

This research explores the tactical use of shortcut path options to allow time recovery for late arrivals. Before human-in-the-loop experiments can be entertained or even prototype shortcut paths can be designed, we need to explore the range of design parameters. This paper presents sensitivity analyses studying the effect of shortcut design parameters and shortcut availability on spacing performance. Additionally, throughput and delay performance is compared between shortcut use policies and scheduling without shortcuts for a wide range of

demand rates. This is followed by a discussion of operational considerations and future steps and then conclusions.

Background

Whereas path-based control in the terminal area is not a new concept, when coupled with scheduling, flights are usually scheduled to the shortest paths, reserving longer paths for exception handling. The following concepts have their own unique benefits but combined together may be more effective in achieving schedule conformance.

Haraldsdottir et. al. conducted a trade study of several arrival management concepts applied to Washington Dulles International Airport, including RNAV and RNP routing with path options [9]. Path stretching options based on “tromboning” and straight-in vectoring techniques were made available to the scheduler to increase the amount of delay the terminal airspace could absorb. However, this use of path options in single stage scheduling tended to increase scheduled delay (due to the granularity that leads to the next largest path option being chosen), and increase the likelihood of separation violations (due to additional crossing paths they create).

Uebbing-Rumke and Temme changed flight paths to segregate unequipped aircraft from those performing Continuous Decent Arrivals (CDAs) [4]. CDA traffic were scheduled to the shortest most direct arrival transition due to their increased ability to predict arrival times. Unequipped aircraft were given longer arrival transitions to the final merge point to give controllers time to merge the aircraft with CDA traffic.

Swenson et. al. applied missed approach paths coupled to the scheduler to more efficiently integrate these flights back into already crowded arrival streams [7]. Whereas this method did not impact throughput or controller workload significantly, the flight paths were more predictable and resulted in fewer and less severe separation violations than without the predefined paths.

Callantine et. al. investigated the use of standard path options as a controller tool to recover from highly disruptive off-nominal events such as emergency landings, missed approaches, and non-communicating aircraft [10]. In this study, most of the path options offered were path stretches with

relatively few shortcuts. The study reported that the charted path options were not always useful to controllers, either because aircraft had already passed suitable options, or controllers did not have time to select and implement them. However, of the two sectors where controllers issued the most path options, the shortcuts were among the most often used.

During the author’s follow-up discussions with Callantine, he postulated that one reason for the heavier use of standard shortcuts than path-stretch options may be that shortcuts more closely mimicked commonly used path control techniques, and it was harder for controllers to visualize how some of the path stretch options reintegrated flights into the standard flow. Another reason may be that there are simply less speed control options to advance a flight than to delay. In addition to maximum speed restrictions, increasing speed in the arrival phase of flight is not well accepted by pilots or controllers and is not always feasible. Callantine et. al. reported that wind shifts that ‘undo’ previous controller actions seemed most frustrating to the controllers, particularly if they created a sudden need to ‘catch up’ to the schedule [10].

The point merge concept [11-13] is different from those mentioned above in that it is not time-based. RNAV routes leading to merge points end with isometric segments tangential to the merge point. As a flight travels the isometric segment, controllers decide when to issue a direct-to the merge point. If the nominal path continues to the end of the RNAV route, the controller direct-to instructions are essentially issuing a shortcut path option. The concept improved spacing precision (reduced spacing standard deviation at the final approach fix) by roughly 20% and reduced the number of instructions per aircraft by roughly 50% [12].

The study presented in this paper shows that a combined approach is beneficial by merging the point-merge path-based approach with the time-based approach by applying tactical shortcut path-options as a means to improve schedule conformance.

Shortcut Time Recovery Concept

Given a single or set of coordination points (e.g. meter fixes, merge points, runway thresholds) within a fixed terminal routing structure, a scheduler may use a nominal time-to-fly for a given engine or

aircraft type along each route segment to generate scheduled times at each coordination point. As errors accumulate along route segments, a flight may adjust its speed either as directed by a ground-based controller or by an advanced on-board flight management system. In either case, the level of precision with which the flight can be controlled to a given coordination point determines the scheduling buffer (extra padding added to required separation) and ultimately the maximum throughput of the coordination point. Thipphavong and Mulfinger [14] studied the relationship between flight arrival uncertainty, scheduling buffer, and controller intervention rate (percentage of flight pairs expected to lose separation without controller intervention). A suitable scheduling buffer can be calculated for a given arrival uncertainty and desired maximum controller intervention rate.

The shortcuts concept presented in this paper assumes that at some point prior to the coordination point, late flights have the opportunity to take a shortcut path option and recover time. This would change the shape of the arrival uncertainty curve and affect the relationship between scheduling buffer size and controller intervention rate. Shortcuts are defined by the amount of time recovery they offer and the policy governing the use of the shortcut. The following subsections define design parameters for single and multiple shortcuts with schedule-based and spacing-based use policies.

Schedule-based Shortcuts

Two parameters define the schedule-based shortcut: late threshold L , and shortcut recovery S . S is the time a flight can expect to recover if it uses the shortcut. All flights that are late more than L use the shortcut, thereby adding S to their arrival time error ε at the coordination point. The modified error of the i^{th} flight becomes

$$\varepsilon'_i = \begin{cases} \varepsilon_i, & \varepsilon_i \geq -L \\ \varepsilon_i + S, & -L > \varepsilon_i \end{cases} \quad (1)$$

where the first and second rows of the equation represent options to not use or to use the shortcut respectively.

Figure 1 (a) identifies flights using the shortcut as the orange portion of a Gaussian error distribution with error less than $-L$. In Figure 1 (b), the orange

portion of the error distribution shifts to the right by S . The new error distribution is the sum of the blue and shifted orange curves in Figure 1 (b). Although this tends to create an average error bias to be early, the variation in error is expected to be smaller, which would allow smaller buffers between scheduling slots.

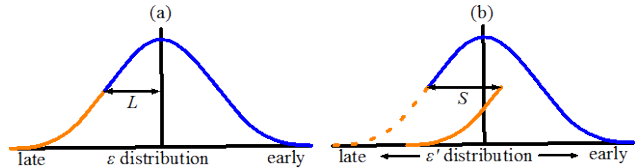


Figure 1. Single Shortcut (a) Late Threshold L and (b) Shortcut Recovery S

More than one shortcut may be specified for a given coordination point. For the purposes of this analysis, S represents the largest shortcut recovery with smaller shortcut recoveries placed at equal intervals between S and 0. Because each shortcut must have a unique late threshold, L now specifies the threshold late error of a flight beyond what the next shortest shortcut enables. For two shortcuts, the modified error of the i^{th} flight becomes

$$\varepsilon'_i = \begin{cases} \varepsilon_i, & \varepsilon_i \geq -L \\ \varepsilon_i + \frac{1}{2}S, & -L > \varepsilon_i \geq -(L + \frac{1}{2}S) \\ \varepsilon_i + S, & -(L + \frac{1}{2}S) > \varepsilon_i \end{cases} \quad (2)$$

where the first, second, and third rows of the equation represent options to not use any shortcut, use the shortcut offering less time recovery ($\frac{1}{2}S$), or use the shortcut offering more time recovery (S) respectively.

The error distribution for two shortcuts is shown in Figure 2. The blue, magenta, and orange portions of the distribution curves represent flights equaling the first, second, and third rows of the above equation.

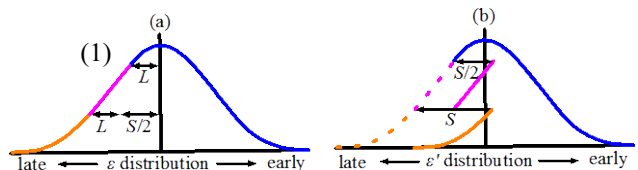


Figure 2. Two Shortcut (a) late Threshold L and (b) Shortcut Recovery S

For three shortcuts, the modified error of the i^{th} flight becomes

$$\mathcal{E}'_i = \begin{cases} \varepsilon_i, & \varepsilon_i \geq -L \\ \varepsilon_i + \frac{1}{3}S, & -L > \varepsilon_i \geq -(L + \frac{1}{3}S) \\ \varepsilon_i + \frac{2}{3}S, & -(L + \frac{1}{3}S) > \varepsilon_i \geq -(L + \frac{2}{3}S) \\ \varepsilon_i + S, & -(L + \frac{2}{3}S) > \varepsilon_i \end{cases} \quad (3)$$

Use of the late threshold parameter bases the decision to take a shortcut entirely on schedule. Therefore, given a required time of arrival at the scheduling point, the decision to take a shortcut could be delegated to ground with the aid of timelines and other schedule conformance aids, or to a flight deck equipped to conduct required time of arrival operations.

Spacing-Based Shortcuts

Another shortcut use policy could base the decision to shortcut on spacing. A flight would use a shortcut only if it results in sufficient separation with the leading flight. Whereas a schedule-based use policy requires information about a single flight, a spacing-based use policy requires information about a pair of sequential flights. This may be a more complicated use policy to delegate to the flight deck, as with flight deck interval management. To simplify the parametric analyses and comparison to schedule-based shortcuts, spacing is expressed as the relative arrival time error between a pair of sequential flights. This relative error is computed after the leading flight has made the decision to use a shortcut or not. For the i^{th} flight and its leading flight ($i-1$), the relative error is $\mathcal{E}'_{i-1} - \varepsilon_i$. Instead of the late threshold, a new relative error threshold parameter, R , defines the acceptable relative error between a leading and trailing flight for the trailing flight to use the shortcut. The single relative spacing shortcut modified error of the i^{th} flight is

$$\mathcal{E}'_i = \begin{cases} \varepsilon_i, & \mathcal{E}'_{i-1} - \varepsilon_i + R < S \\ \varepsilon_i + S, & S \leq \mathcal{E}'_{i-1} - \varepsilon_i + R \end{cases} \quad (4)$$

For two relative spacing shortcuts, the modified error of the i^{th} flight is

$$\mathcal{E}'_i = \begin{cases} \varepsilon_i, & \mathcal{E}'_{i-1} - \varepsilon_i + R < \frac{1}{2}S \\ \varepsilon_i + \frac{1}{2}S, & \frac{1}{2}S \leq \mathcal{E}'_{i-1} - \varepsilon_i + R < S \\ \varepsilon_i + S, & S \leq \mathcal{E}'_{i-1} - \varepsilon_i + R \end{cases} \quad (5)$$

For three relative spacing shortcuts, the modified error of the i^{th} flight is

$$\mathcal{E}'_i = \begin{cases} \varepsilon_i, & \mathcal{E}'_{i-1} - \varepsilon_i + R < \frac{1}{3}S \\ \varepsilon_i + \frac{1}{3}S, & \frac{1}{3}S \leq \mathcal{E}'_{i-1} - \varepsilon_i + R < \frac{2}{3}S \\ \varepsilon_i + \frac{2}{3}S, & \frac{2}{3}S \leq \mathcal{E}'_{i-1} - \varepsilon_i + R < S \\ \varepsilon_i + S, & S \leq \mathcal{E}'_{i-1} - \varepsilon_i + R \end{cases} \quad (6)$$

Multiple Shortcut Analysis

Sensitivity analyses were performed by varying the shortcut design parameters and analyzing their effects on aircraft spacing at a single scheduling point. This was done to gain insight into how to design shortcuts into a terminal route structure to maximize the benefit.

Method

Analyses presented in this and following sections are based on a Gaussian distribution of $\varepsilon=0+/-\sigma$ to model aircraft arrival time errors at the scheduling point. The shortcut design parameters are designed to scale to a given expected σ such that $S=\alpha_S\sigma$, $L=\alpha_L\sigma$, and $R=\alpha_R\sigma$. Sensitivity analyses vary α_S , α_L , and α_R to produce separation results that also scale with σ .

A Gaussian distribution of $0+/-1$ was sampled $N=10,000$ times to represent a Monte Carlo sequence of aircraft arrival time errors at a single scheduling point. The errors were modified based on the shortcut design parameters. The late threshold α_L or relative error threshold α_R was varied from 0 to 2 and the shortcut recovery α_S was varied from 0 to 3.

Let \mathcal{E}'_i and \mathcal{E}'_{i-1} be the performance error of a trailing and leading aircraft, respectively, in the Monte Carlo sequence. The separation error between the aircraft is $E = \mathcal{E}'_i - \mathcal{E}'_{i-1}$. Assume the aircraft are scheduled to arrival slots separated by their minimum required separation plus a buffer B . If $E > B$, then controller intervention is required to maintain separation. Therefore, the x^{th} percentile E in the Monte Carlo sequence is also the minimum required B to stay below a 100-x% controller intervention rate. This relationship is scalable with σ such that $E=\alpha_E\sigma$, $B=\alpha_B\sigma$, and $\alpha_E > \alpha_B$ requires controller intervention.

Schedule-Based Shortcut Results

Figure 3 shows single shortcut area plots of 90, 80, 70, and 60th percentile α_E for ranges of α_L and α_S . These α_E percentiles translate to the α_B required for 10%, 20%, 30%, and 40% controller intervention rates.

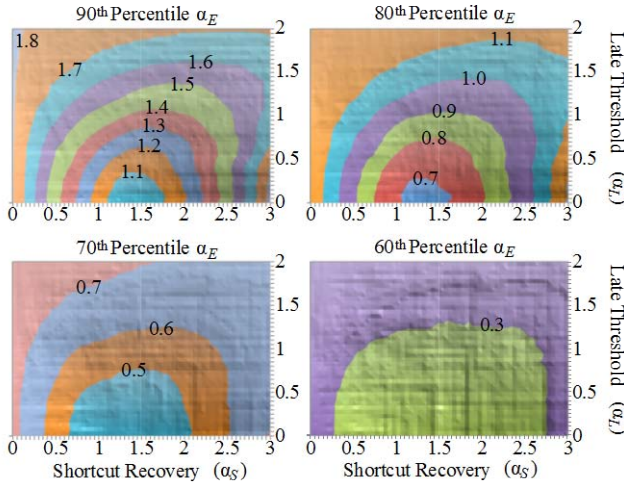


Figure 3. Area Plots of Spacing Error for Single Schedule-Based Shortcut Parameters

Each distinct colored shaded area represents a 0.1 range of spacing error results bounded by the values labeling their borders. Shortcut recovery increases along the x-axis and the late threshold increases along the y-axis. The spacing errors at $(\alpha_S, \alpha_L)=(0,0)$ represent a scheduling point with no shortcut. Figure 3 may be used as a lookup table to inform shortcut design and scheduling buffer settings. For example, assume one wanted to augment an existing terminal route with a single shortcut leveraging existing fixes and a scheduler allowing no more than 20% separation violation probability impacting controller workload. Assume the route geometry allows a design choice between a shortcut with $\alpha_S = 0.5$ or 1. The 80th percentile graph (accommodating the <20% controller intervention rate requirement) shows that minimum errors are ~0.9 or ~0.7 for the $\alpha_S = 0.5$ or 1 shortcuts respectively when $\alpha_L=0$. Therefore, the best choice would be to implement the $(\alpha_S, \alpha_L)=(1,0)$ shortcut with at least a 0.7 scheduling buffer.

At lower error percentiles (i.e. higher controller intervention rates), errors (i.e. required buffers) are lower and less sensitive to the shortcut parameters. The lowest errors tend to result for a mid range of α_S

and $\alpha_L=0$. Only when α_S is greater than 2.5, increasing α_L to approximately one half α_S reduces the error slightly.

The same analysis was completed for a scheduling point with two and three shortcuts. For each set of shortcuts, the largest shortcut recovery parameter α_S was varied from 0 to 3. The late threshold α_L (now relative to the next shortest shortcut) was varied from 0 to 2. For these ranges of α_S and α_L , $\alpha_L=0$ consistently produced the lowest spacing errors. Figure 4 compares the minimum spacing errors (the minimum α_E achievable for a given percentile) between one, two, and three shortcut scheduling points. The no shortcut errors for $(\alpha_S, \alpha_L)=(0,0)$ are also shown.

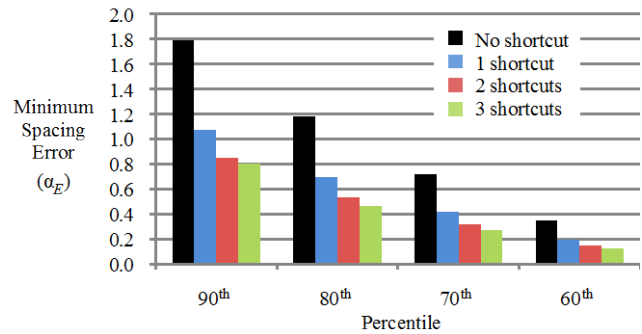


Figure 4. Minimum Spacing Error for 1, 2, and 3 Schedule-Based Shortcuts

As seen in Figure 4, the potential improvement over no shortcuts is impressive. The 90th percentile error (i.e. minimum scheduling buffer required to achieve less than 10% controller intervention) could be reduced roughly 35-55% depending on whether a one or two shortcut design is used. Reducing the scheduling buffers would enable increased throughput at the coordination point. Or if throughput is not an issue and rather reducing controller workload is desired, the probability of separation violation could be reduced 15-20% depending on whether a one or two shortcut design is used.

Whereas the minimum spacing error gets smaller for each additional shortcut option, the reduction is less pronounced from two to three shortcuts than from one to two. This is an important insight to consider if future human-in-the-loop studies determine that managing additional shortcut options affect controller workload. It may not be worth it to reduce controller workload associated

with maintaining separation if it is only replaced by workload associated with managing shortcut options. The diminishing benefit of additional shortcuts may be because the range of shortcut recovery options begins to exceed the range of flights late enough to use them beyond two shortcuts. Multiple shortcuts were placed at equal intervals between S and 0 . A different distribution of shortcuts could produce different results.

Figure 5 compares one, two, and three shortcut α_E for the full range of α_S when $\alpha_L=0$. Spacing errors within 0.05 of the minimum errors are shown in Figure 5 with a diamond shape to identify the range of optimal α_S producing close to minimum errors. This shows error sensitivity to shortcut recovery α_S .

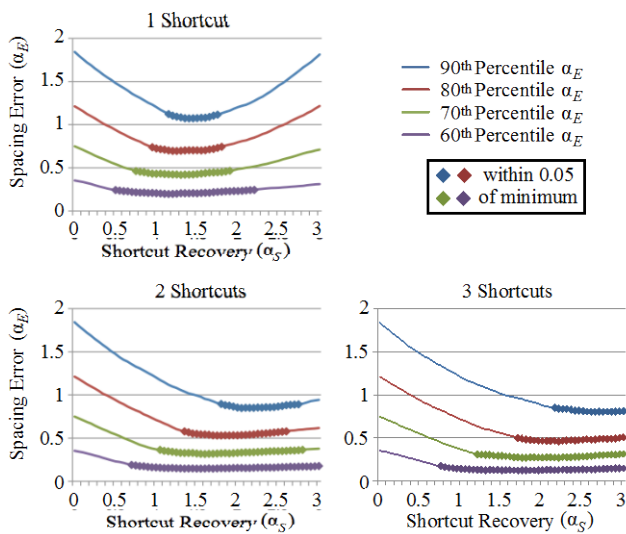


Figure 5. Spacing Error vs. Shortcut Recovery for Schedule-Based Shortcuts

Although a shortcut may be designed to a nominal speed and time recovery, individual flight time will vary. Therefore, lower sensitivity or a wider range of α_S is preferable. The single shortcut 90th percentile optimal α_S range is between 1.15 and 1.75. This means that if a shortcut was designed to be $\alpha_S=1.45$ based on nominal flight time, it could tolerate +/- 20% flight time deviations and still stay below a ~10% controller intervention rate with a buffer as low as $\alpha_B=1.1$.

There are wider ranges of optimal α_S at lower percentiles. The range widens further as more shortcut recovery options are available. With more shortcuts, the optimal α_S tends to be higher which can be seen as close to minimum α_S shift to the right from

one to two to three shortcuts. However, this is expected because the one or two additional shortcuts at equal intervals below α_S still tend to result in an average shortcut recovery in the mid range.

Spacing-Based Shortcut Results

Figure 6 shows single shortcut area plots of 90, 80, 70, and 60th percentile α_E for ranges of α_R and α_S . Unlike α_L in Figure 3, the lowest spacing errors in Figure 6 do not occur at $\alpha_R=0$. In Figure 6, black line curves identify a distinct valley in each area surface. These curves track the α_R values producing the minimum spacing errors as α_S is varied. The minimum error occurs where α_R is smallest along these curves, which become flatter (i.e. less sensitive to α_S) at lower error percentiles. This trend holds for two and three shortcuts as well.

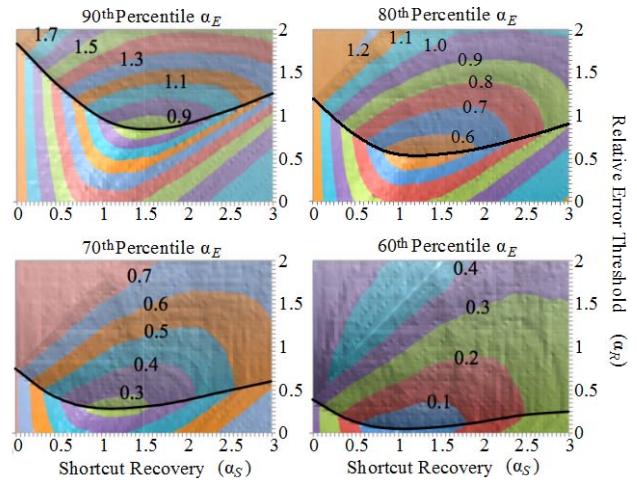


Figure 6. Area Plots of Spacing Error for Single Spacing-Based Shortcut

Figure 7 compares the minimum schedule-based α_E (from Figure 4) and spacing-based α_E for one, two, and three shortcuts. The spacing-based method produces smaller minimum α_E than the schedule-based method. The difference in α_E between the methods increases as the number of shortcuts increase. The 90th percentile error (i.e. minimum scheduling buffer required to achieve less than 10% controller intervention) could be reduced roughly 55-72% or the probability of separation violation could be reduced 20-25% relative to no shortcuts depending on whether a one or two shortcut design is used. This means spacing-based shortcuts can reduce buffer size an extra ~20% or reduce separation violation probability an extra ~5% over schedule-based

shortcuts. The additional benefit still diminishes as more spacing-based shortcut options are added, but not as severely as with schedule-based shortcuts.

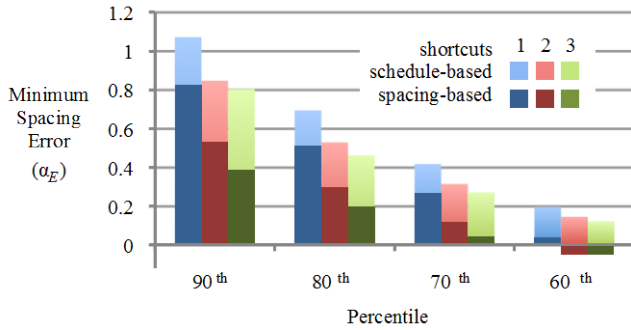


Figure 7. Schedule-Based vs. Spacing-Based Minimum Spacing Error

Figure 8 shows minimum α_E vs. α_S and Figure 9 shows minimum α_E vs. α_R . Values of α_E within 0.05 of the minimums shown in Figure 7 are indicated with a diamond to show the ranges of α_S and α_R producing close to minimum errors. The α_E vs. α_S plots in Figure 8 show similar trends to those in Figure 5 except that changes to α_E sensitivity (i.e. span of close to minimum errors) to percentile and α_S are much less pronounced.

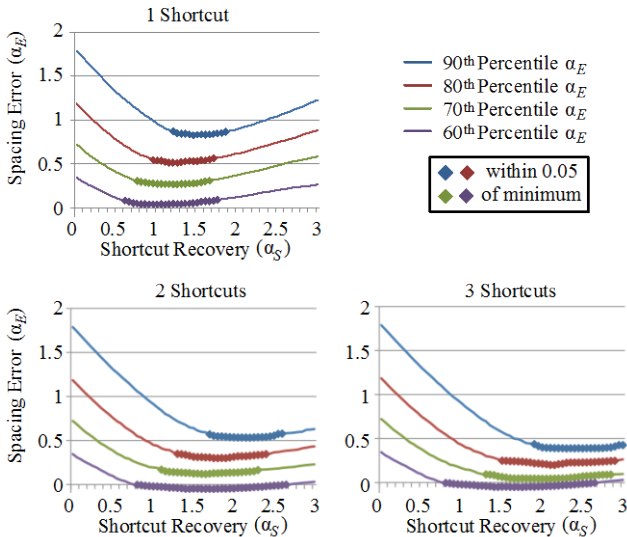


Figure 8. Spacing Error vs. Shortcut Recovery for Spacing-Based Shortcuts

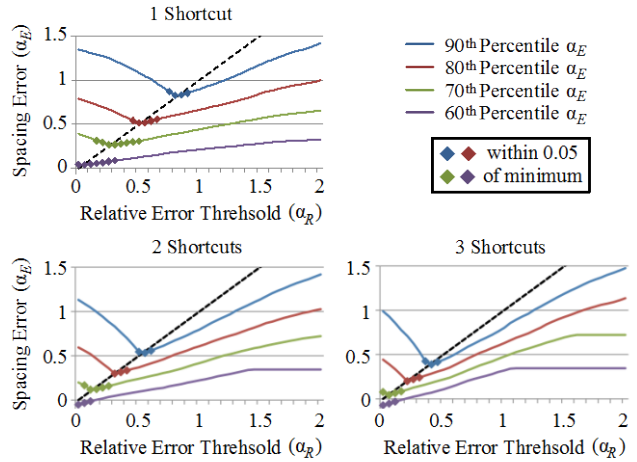


Figure 9. Spacing Error vs. Relative Error Threshold for Spacing-Based Shortcuts

All the α_E vs. α_R plots in Figure 9 show very sharp minima where α_E is equal to α_R . The black dotted line marks $\alpha_E = \alpha_R$ which clearly intersects all the error minima. Assuming $\alpha_E = \alpha_B$ in order to achieve the 100%-percentile controller intervention rate, then $\alpha_R = \alpha_B$ uses the shortcuts to get trailing aircraft as close to the leader as possible without losing separation, essentially removing the buffer. The error sensitivity to α_R is generally much higher for α_S and the sensitivity tends to increase with more shortcuts, rather than decrease as it does with α_S . This means that the spacing-based use policy's low error hinges upon its ability to remove the buffer and space flights as close to the minimum required separation as possible. Whereas this may be desired at a final scheduling point like the runway threshold, it may not be desirable for an upstream scheduling point to remove all the scheduled buffer for any downstream scheduling points. For this reason, schedule-based shortcuts may be preferable farther upstream within a multipoint scheduling network, and spacing-based shortcuts may be preferable farther downstream.

Shortcut Availability Analysis

The geometry of merging routes may not offer all flights the same shortcut time recovery options. When one route joins another straight route, only the joining route may have shortcut opportunities. A sensitivity analysis was performed to determine if the percentage of traffic with available shortcuts has an effect on the desired range of shortcut design parameters (α_S , α_L , and α_R).

The percentage of flights with available shortcuts was varied from 0% to 100% in 10% increments. The 90th percentile separation errors were analyzed for schedule-based shortcuts and spacing-based shortcuts assuming a desired maximum probability of separation violation (i.e. controller intervention rate) of 10%. Figures 10 and 11 show minimum 90th percentile spacing error sensitivity to schedule-based shortcut parameters. Sensitivity to shortcut recovery α_S is shown in Figure 10 and sensitivity to late threshold α_L is shown Figure 11. Purple diamonds represent optimal parameter values (values producing the minimum spacing error α_E) with whiskers identifying the optimal range (parameter values producing α_E within 0.05 of the minimum). The minimum α_E values are shown in blue on the same scale for easy comparison. This is a more compact way of viewing results similar to those shown in Figures 5, 8 and 9. Note that the same single shortcut 90th percentile optimal α_S range between 1.15 and 1.75 show in Figure 5 is shown in Figure 10 in the 100% shortcut availability column.

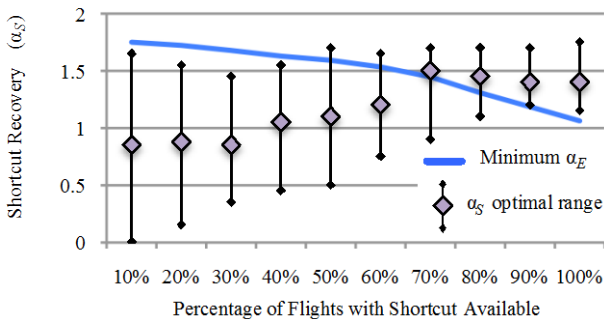


Figure 10. Spacing Error Sensitivity to Shortcut Recovery for Varying Schedule-Based Shortcut Availability

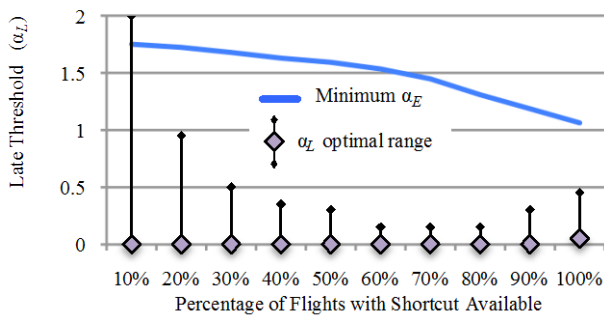


Figure 11. Spacing Error Sensitivity to Late Threshold for Varying Schedule-Based Shortcut Availability

As shortcut availability increases from 0% to 100%, the optimal ranges of α_S get narrower (i.e. α_E gets more sensitive to α_S). Although optimal α_S values tend to increase as more flights are able to use the shortcut, the optimal ranges overlap such that α_S values between 1.2 and 1.45 produce close to minimum α_E regardless of the shortcut availability. This means that even if shortcut availability is expected to vary widely at a given coordination point (for example, irregular flows of traffic arriving from a straight path or merging path), a single design choice for the recovery time the shortcut offers will suffice. The α_L optimal ranges vary as the shortcuts availability changes. However, the optimal α_L values remain stable at $\alpha_L=0$. This reinforces the insight that all flights expected to be late by any amount should use the shortcut. However, the minimum spacing error α_E , and therefore minimum required scheduling buffer does change (between 1.8 and 1.1) with shortcut availability. The scheduler must then be able to adapt to expected traffic demand ratios and insert the appropriate buffer. Otherwise, a constant buffer large enough to accommodate the lowest expected shortcut availability should be used.

Figures 12 and 13 show 90th percentile spacing error sensitivity to spacing-based shortcut parameters. As with schedule-based shortcuts, the minimum spacing error α_E values for spacing-based shortcuts also vary (between 1.8 and 0.8) with shortcut availability requiring adaptable scheduling buffers. However, the effect of shortcut availability on optimal parameters is quite different with spacing-based shortcuts than with schedule-based shortcuts. As seen in Figure 12, the optimal ranges for α_S are relatively large creating an overlapping optimal range between 1.25 and 1.9, which is 2.6 times larger than the overlapping optimal range for schedule-based shortcuts. The optimal α_S values themselves are quite stable, varying between 1.4 and 1.7. This wide range of acceptable α_S opens up the shortcut design flexibility to accommodate specific route geometry or other operational constraints.

Unlike the late threshold α_L , the relative error threshold α_R is very sensitive to shortcut availability. As seen in Figure 13, the optimal α_R values consistently equal the minimum α_E as they decrease with increased shortcut availability. For the spacing-based use policy, the relative error threshold R should

always equal the scheduling buffer B . Simply put, the spacing-based use policy achieves minimum spacing error by always using the shortcut unless it would violate required separation with the leading flight. This policy does not change with shortcut availability.

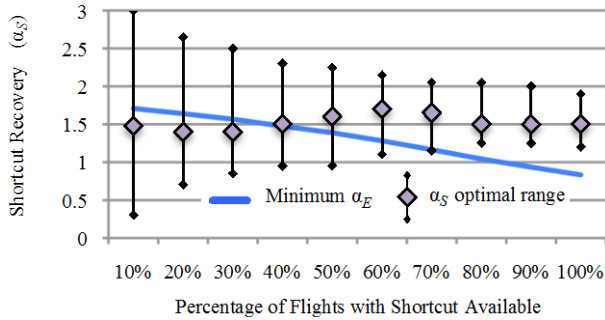


Figure 12. Spacing Error Sensitivity to Shortcut Recovery for Varying Spacing-Based Shortcut Availability

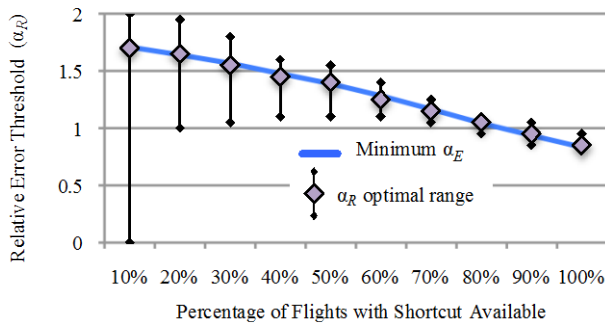


Figure 13. Spacing Error Sensitivity to Relative Error Threshold for Varying Spacing-Based Shortcut Availability

Demand Analysis

The previous section demonstrated how lower spacing errors are achieved and therefore smaller scheduling buffers may be used when shortcut availability is higher. Routing may be redesigned to ensure that there are no merges that limit shortcuts availability. This requires bending and lengthening the straight route such that the required shortcut is available. However, this transformation only highlights the fact that the very existence of shortcuts implies that the scheduled route is not the shortest. The main advantage of reserving shortcuts as tactical path options is to increase throughput. So when demand is low and high throughput is not needed, it

may be better to schedule to the shorter path to save flight time and fuel.

This section analyzes the use of tactical shortcuts vs. scheduling to the shortest path for a range of demand conditions. Because throughput is ultimately constrained by the minimum required separation, experiment variables and results were expressed as functions of a generic required separation $ReqSep$ that could be expressed as time or distance.

Method

Streams of 100 flights were simulated at demand rates varying from 0 to 1 flights per $ReqSep$ in increments of 0.01 flights per $ReqSep$. A flight's Estimated Time of Arrival (ETA) was modeled as a uniformly distributed random value between 0 and $100/ReqSep$.

The flights were then sequenced by ETA and Scheduled Times of Arrival (STAs) were assigned to space flights by at least $ReqSep$ plus a scheduling buffer. The scheduling buffer was determined by the shortcut use policy and a given σ of the expected error distribution. A baseline σ was defined as $0.2ReqSep$. This definition was based on reference [1], which observed ~ 0.6 nmi standard deviation of arrival spacing for flight pairs requiring 3 nmi separation ($\sigma=0.6=0.2*3=0.2ReqSep$).

A Gaussian error $\varepsilon=0\pm\sigma$ was added to the STA to get an original Actual Time of Arrival (ATA_{orig}). For tactical shortcut simulations, a schedule-based or spacing-based use policy could modify ATA_{orig} by subtracting S to get the final Actual Time of Arrival (ATA), where $S=1.5\sigma=0.3ReqSep$.

Four shortcut use policies were employed to modify ATA_{orig} by choosing to use or not use a shortcut resulting in a final ATA. All buffers were designed to require no more than 10% controller intervention in avoiding separation violations. The baseline "No Shortcut" policy used buffer $B=1.8\sigma$ and shortcut recovery $S=0$ signifying that the shortest path was used for scheduling and there was no tactical shortcut available. Schedule-based and spacing-based use policies were also analyzed. The schedule-based policy used buffer $B=1.1\sigma$ to accommodate single shortcut parameters $S=1.5\sigma$ and $L=0$. The spacing-based policy used buffer $B=0.9\sigma$ to accommodate parameters $S=1.5\sigma$ and $R=B$. In

addition, a hybrid method utilizing both schedule-based shortcuts and “No Shortcut” scheduling was analyzed. This method scheduled flights to either the shorter or longer path depending on expected need for precision. Let $B_1=1.8\sigma$ be the “No Shortcut” buffer and let $B_2=1.1\sigma$ be the schedule-based shortcut buffer. If the longer path ETA difference between a trailing and leading flight was at least B_1 , the flight ETA would be advanced by $S=1.5\sigma$ and scheduled to the shorter path using B_1 as the buffer. Otherwise, the flight was scheduled to the longer path using B_2 with schedule-based shortcut parameters $S=1.5\sigma$ and $L=0$.

Three metrics were collected for each 100-flight simulation. These were the percentage of flights using the shortcut, throughput, and average flight delay. Throughput was defined by the ATA make span divided by 100 flights (flights per *ReqSep*). Individual flight delay was defined as the sum of scheduled delay (STA-ETA) and a path delay of S if the longer path was used. Delay attributed to ϵ was not included as it would average to 0. The individual flight delay was summed and then divided by 100 to get average flight delay.

Simulations of 100 flights were repeated 1,000 times for each demand rate and shortcut use policy. The mean and standard deviation for each of the three metrics are displayed in the results.

Results

Figure 14 shows percentage of flights using the shortcut for each use policy as demand increases from 0 to 1 aircraft per *ReqSep*. Only the three use policies involving tactical shortcuts are shown because “No Shortcut” has 100% shortcut use for any demand because it schedules to the shortcut path in the first place. Thick lines represent mean results for the 1,000 repetitions of each demand and use policy. Thinner lines bounding the mean results represent mean +/- standard deviation for the 1,000 repetitions.

Schedule-based shortcuts are used ~50% of the time regardless of demand. This is because the decision to use the shortcut is driven entirely by flight arrival time error ϵ , which is modeled independent of demand. The Gaussian error model $\epsilon=0\pm\sigma$ is greater than $L=0$ for ~50% of the time resulting in ~50% shortcut use.

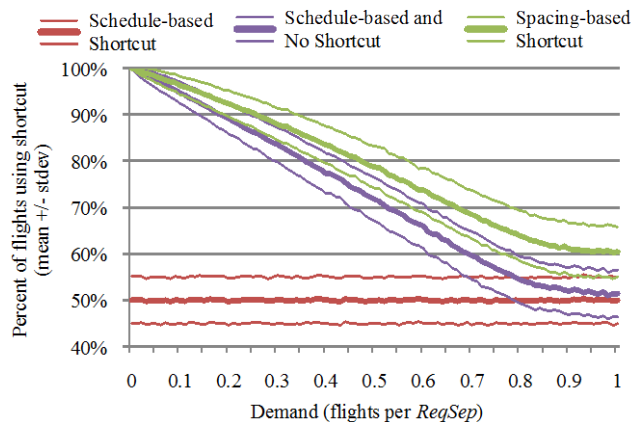


Figure 14. Percent of Flights Using Shortcut vs. Demand

The other two shortcut usage policies are influenced by demand. The spacing-based method always directs flights to use the shortcut unless required separation with leading flights would be violated. Therefore, shortcuts are used more often when demand is lower. A similar trend occurs with the hybrid schedule-based and “No Shortcut” policy. This policy operates similar to spacing-based only it does it more strategically by deciding to schedule a flight to the shorter path with a larger buffer, or to the longer path with a smaller buffer. The flights scheduled to the shorter path used the shortcut path 100% of the time and the flights scheduled to the longer path used the shortcut 50% of the time. Because more flights are scheduled to the shorter path when demand is lower, the percentage of flights using the shortcut follows the same trend. Shortcut usage is directly related to path delay. So these results show that for lower demand the spacing-based and hybrid use policies incur the least path delay.

Figure 15 shows mean throughput results for each use policy as demand increases. Results for “No Shortcut” are shown in addition to the three use policies from Figure 14. Thinner lines representing mean +/- standard deviation are shown for only “No Shortcut” to minimize graph clutter. All use policies display a similar trend of increasing throughput with demand up to a saturation point, where the throughput curve levels off. The saturated throughput of each use policy is inversely related to the scheduling buffer size. “No Shortcut” saturates at the lowest throughput because it uses the largest scheduling buffer. Spacing-based shortcut saturates at the highest throughput (~14% higher than No Shortcut) because it uses the smallest scheduling

buffer. The schedule-based and hybrid (schedule-based and “No Shortcut”) methods have almost identical throughput vs. demand behavior (both saturating ~11% higher than “No Shortcut”). This is because as demand increases, the hybrid method schedules more and more flights similar to the schedule-based method. By the time the hybrid method saturates, most flights are scheduled to the longer path like the schedule-based method.

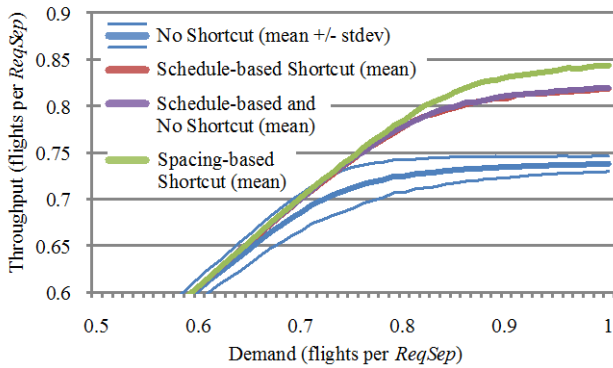


Figure 15. Throughput vs. Demand

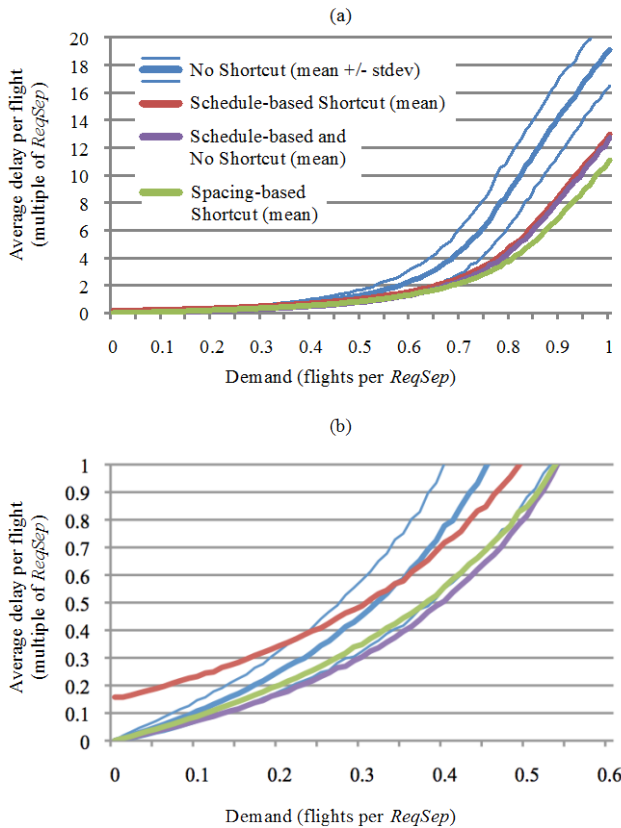


Figure 16. Average Delay vs. Demand

Figure 16 shows average flight delay results for each use policy as demand increases. As seen in Figure 16 (a), all use policies have low average flight delay at low demand. Then as demand begins to saturate the schedule, the delay begins to rapidly increase. The rate of delay increase after saturation is directly related to schedule buffer size. Fig 16 (b) is scaled to show how the average delay differs between use policies at lower demand levels when path delay dominates rather than scheduled delay. The schedule-based shortcut average delay never reaches zero because consistently 50% of flights incur path delay. Whereas the lower buffer enables schedule-based shortcuts to have less scheduling delay than “No Shortcut” at higher demands, when the demand gets low enough, the inherent path delay associated with schedule-based shortcuts exceeds the schedule delay difference. The hybrid schedule-based and “No Shortcut” policy retains the benefit of schedule-based shortcuts at higher demands and the benefit of “No Shortcut” at lower demands that rivals spacing-based shortcuts.

These results show that there is a great advantage to integrating shortcut use policies and trading off between scheduling to the shorter path to minimize delay and scheduling to the longer path to maximize throughput. This is a powerful insight into the design of a sophisticated scheduler that can tailor its choice of path and scheduling buffer for an individual flight to meet the traffic situation.

Discussion and Future Work

The above analyses show the potential tactical shortcuts have to increase throughput and reduce delay. However, some operational considerations must be explored.

The tactical shortcut concept is expected to be most useful under conditions when speed-based time recovery is restricted. Procedural constraints, such as the 250 knot maximum within the terminal airspace, limit the range of feasible time recovery. The performance characteristics of an aircraft at given altitudes add more constraints. Errors can accumulate quickly and unexpectedly, too close to the scheduling point to recover with speed. Or a flight may be limited by the minimum time required to descend to a maximum altitude constraint. The first three conditions mentioned are limitations of speed that shortcut path can address. However, the last

condition is limited by time, which neither increasing speed nor shortening the flight path can address. Flights following optimal descent profiles scheduled to a longer path may not be able to utilize a shortcut. For tactical shortcuts to be used, a flight would need to follow a less optimal descent profile with respect to its scheduled path.

If tactical shortcuts are to be used to reduce controller intervention rate for a given scheduling buffer, workload associated with identifying if a flight should take a shortcut and issuing the shortcut command to the flight needs to be explored. This may depend heavily on the specific shortcut geometry and decision support tools available to the controller or pilot.

The analyses presented in this paper deal only with a single scheduling point. Future analysis should apply the concept to multi-point scheduling. Spacing-based shortcuts outperformed the rest by removing as much scheduled buffer as it could by matching R with B . Whereas this may be desired at a final scheduling point like the runway threshold, it may not be desirable for an upstream scheduling point to remove all the scheduled buffer for any downstream scheduling points.

Finally, future analysis should explore integration of aircraft with mixed arrival time precision. For example, flights performing flight-deck interval management or flights capable of achieving high precision required times of arrival may be scheduled differently than less equipped flights that require more ground support. Ideally, a scheduler should be able to integrate these flights while taking advantage of their unique capabilities. A hybrid approach similar to the one demonstrated in the previous section may be used to schedule the appropriate sized buffer and path to flights depending on their speed-based arrival time precision capabilities.

Conclusion

This paper introduced a concept using tactical shortcuts to improve arrival precision and thereby increase throughput. In general, shortcuts should be designed to recover time approximately 1.5 times the standard deviation of expected arrival time performance at the coordination point. Spacing error can be reduced more as more shortcut options are designed. A spacing-based shortcut use policy

outperformed a schedule-based use policy. Schedule-based shortcuts can potentially reduce scheduling buffer size by 35-55% or separation violation probability by 15-20%. Spacing-based shortcuts can potentially reduce scheduling buffer an additional 20% or separation violation probability an additional 5%. It was determined in an analysis of variable shortcut availability that for both methods, a single shortcut length as defined by the time it is designed to recover will suffice. However, the scheduling buffer would still need to either adapt to the expected shortcut availability or be set to accommodate the minimum expected shortcut availability. Throughput and delay performance of each shortcut use policy was measured for a wide range of demand rates. Whereas spacing-based shortcuts performed the best, this performance was due to removing much of the scheduled buffer, which may not be beneficial for downstream scheduling points in a multi-point scheduling system. A use policy that combined the schedule-based shortcut method and scheduling to the shorter path without shortcuts outperformed using either of these methods individually. It also performed as well as the spacing-based use policy when demand was below saturation levels.

Future study should include application to multi-point scheduling, mixed aircraft performance, and operational considerations. For example, it is still unclear what operational workload issuing shortcut path options would add to controllers and what decision support aids would be required. The potential benefits of shortcut time recovery to increase throughput and reduce delay certainly warrant these future studies.

The insights presented in this paper not only inform how to design more path options into existing terminal area routing, but how to design sophisticated schedulers to best take advantage of the additional path options. These schedulers may tailor path and buffer size to individual flights to best accommodate both the flight's individual precision capabilities and the larger traffic situation.

References

- [1] Zelinski, S., 2012, "Benefits of Precision Scheduling and Spacing for Arrival Operations," 31st Digital Avionics Systems Conference, Williamsburg, Virginia, IEEE.

- [2] Korn, B., H. Helmke, and A. Kuenz, 2006, "4D Trajectory Management in the Extended TMA: Coupling AMAN and 4D FMS for Optimized Approach Trajectories," 25th International Congress of the Aeronautical Sciences, Hanburg, Germany.
- [3] Helmke, H., R. Hann, M. Uebbing-Rumke, D. Muller, and D. Wittkowski, 2009, "Time-Based Arrival Management for Dual Threshold Operation and Continuous Decent Approaches," 8th USA/Europe Air Traffic Management R&D Seminar, Napa Valley, California, FAA and EUROCONTROL.
- [4] Uebbing-Rumke, M. and M. Temme, 2011, "Controller Aids for Integrating Negotiated Continuous Decent Approaches into Conventional Landing Traffic," 9th USA/Europe Air Traffic Management R&D Seminar, Berlin, Germany, FAA and EUROCONTROL.
- [5] Swenson, H., J. Thipphavong, A. Sadovsky, L. Chen, C. Sullivan, and L. Martin, 2011, "Design and Evaluation of the Terminal Area Precision Scheduling and Spacing System," 9th USA/Europe Air Traffic Management R&D Seminar, Berlin, Germany, FAA and EUROCONTROL.
- [6] Thipphavong, J., H. Swenson, L. Martin, P. Lin, and J. Nguyen, 2012, "Evaluation of the Terminal Precision Scheduling and Spacing System for Near-Term NAS Application," 1st International Conference on Applied Human Factors and Transportation, San Francisco, California, Amitran.
- [7] Swenson, H., J. Jung, J. Thipphavong, L. Chen, L. Martin, and J. Nguyen, 2012, "Development and Evaluation of the Terminal Precision Scheduling and Spacing System for Off-Nominal Condition Operations," 31st Digital Avionics Systems Conference, Williamsburg, Virginia, IEEE.
- [8] Kupfer, M., T. Callantine, L. Martin, J. Mercer, and E. Palmer, 2011, "Controller Support Tools for Schedule-Based Terminal-Area Operations," 9th USA/Europe Air Traffic Management R&D Seminar, Berlin, Germany, FAA and EUROCONTROL.
- [9] Haraldsdottir, A., J. Scharl, J. King, and M. E. Berge, 2009, "Arrival Management Architecture and Performance Analysis with Advanced Automation and Avionics Capabilities," 9th Aviation Technology, Integration and Operations (ATIO) Conference, Hilton Head, South Carolina, AIAA.
- [10] Callantine, T. J., C. Cabrall, M. Kupfer, L. Martin, J. Mercer, E. A. Palmer, 2011, "Investigating the Impact of Off-Nominal Events on High-Density 'Green' Arrivals," 30th Digital Avionics Systems Conference, Seattle, Washington, IEEE.
- [11] Boursier, L., B. Favennec, E. Hoffman, A. Trzmiel, F. Vergne, and K. Zeghal, 2007, "Merge Arrival Flows without Heading Instructions," 7th USA/Europe Air Traffic Management R&D Seminar, July, Barcelona, Spain, FAA and EUROCONTROL.
- [12] Favennec, B., E. Hoffman, A. Trzmiel, F. Vergne, and K. Zeghal, 2009, "The Point Merge Arrival Flow Integration Technique: Towards More Complex Environments and Advanced Continuous Descent," 9th Aviation Technology, Integration and Operations (ATIO) Conference, Hilton head, South Carolina, AIAA.
- [13] Ivanescu, D., C. Shaw, C. Tamvaclis, and T. Kettunen, 2009, "Models of Air Traffic Management Techniques: Evaluating Performance of Point merge," 9th Aviation Technology, Integration and Operations (ATIO) Conference, Hilton head, South Carolina, AIAA.
- [14] Thipphavong, J. and D. Mulfinger, 2010, "Design Considerations for a New Terminal Area Arrival Scheduler," 10th Aviation Technology, Integration and Operations (ATIO) Conference, Fort Worth, Texas, AIAA.

Acknowledgements

This work was funded by the Concept and Technology Development Project, which is part of NASA's Airspace Systems Program.

Email Addresses

Shannon.J.Zelinski@nasa.gov

*32nd Digital Avionics Systems Conference
October 6-10, 2013*