Title
Improved Clustering for Route-Based Eulerian Air Traffic Modeling

Journal
Journal of Guidance Control and Dynamics, 42(5)

ISSN
0731-5090

Authors
Bombelli, Alessandro
Torné, Adrià Segarra
Trumbauer, Eric
et al.

Publication Date
2019-05-01

DOI
10.2514/1.g003939

Peer reviewed

Improved Clustering for Route Based Eulerian Air Traffic Modeling

Alessandro Bombelli,* Adrià Segarra Torné,† Eric Trumbauer,‡ and Kenneth D. Mease§
University of California Irvine, Irvine, CA, 92697

A new approach is developed for identifying and approximating well-traveled routes in a historical dataset of flight trajectories. The approximate routes are intended for use in a route-based Eulerian model of air traffic flow for strategic planning, but are useful for other route-based strategic planning. The approach involves coarse-clustering, outlier detection, fine-clustering, and aggregate route construction. The coarse-clustering is based on common origin, destination, and average cruise speed. Fine-clustering, based on the Fréchet distance between pairs of trajectories, is applied to each coarse-cluster to subdivide it, if appropriate. The coarse-clustering step reduces the number of trajectory pairs for which the Fréchet distance must be computed. The number of fine-clusters is automatically determined using a combination of three performance indices. Outliers are identified using previously developed methods. The outliers could be discarded or assessed to identify potential routes for avoiding areas that are flight constrained. The effectiveness of the approach for determining aggregate well-traveled routes is demonstrated on a historical dataset for a domain composed of six centers with a total of 19 airports.

Nomenclature

$\mathcal{P}$	=	planning domain
O	=	origin airport
D	=	destination airport
$h(A, B)$	=	haversine distance between points A and B [km]
n_a	=	number of airports in the planning domain
$\underline{d}$	=	pruning distance lower bound [km]
$\overline{d}$	=	pruning distance upper bound [km]

*Ph.D. Student, Department of Mechanical and Aerospace Engineering, abombell@uci.edu.
†Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, adriasegarra@gmail.com.
‡Postdoctoral Researcher, Department of Mechanical and Aerospace Engineering, etrumbau@uci.edu
§Professor, Department of Mechanical and Aerospace Engineering, kmease@uci.edu.

I. Introduction

Air traffic flow management for the National Airspace System (NAS) is currently separated into strategic planning at the national or regional level with a time horizon of 2-8 hours and tactical control which is more local in both space and time. Strategic planning, referred to as Traffic Flow Management (TFM), should anticipate airspace demand exceeding the capacity of airports and sectors, and determine how to adjust the flow to achieve safe operations while limiting delays. The control actions considered at the strategic level are ground delays and pre-departure reroutes. The NAS is physically divided into 22 (20 in the continental US) Air Route Traffic Control Centers. Each center is further divided into sectors, with one or more air traffic controllers responsible for each sector at any given time. In current practice, NAS strategic planning is coordinated from the Air Traffic Control System Command Center and is performed in a committee meeting via telecon every two hours during the day [1].

Model-based optimization approaches have been, and continue to be, developed to assist human decision-makers in strategic planning. Two fundamentally distinct traffic flow modeling approaches derived from fluid mechanics are the Lagrangian approach of tracking each aircraft in the flow [2–4] and the Eulerian approach of tracking the distribution of aircraft over spatially fixed cells in the airspace [5, 6]. Lagrangian models allow aircraft-specific information to be accounted for in planning and are generally high-dimensional and computationally demanding. With the Eulerian approach, aircraft specific decisions are delegated to lower levels of traffic management. Rather than covering the entire airspace with cells*, route-based Eulerian models [7–9] have been pursued. The assumption behind such models is that air traffic flow can be approximated, with accuracy sufficient for strategic planning, by a finite set of routes. Placing the Eulerian cells only on these routes limits the model dimension.

The objective of the present paper is to determine aircraft routes for use in strategic planning. We develop a data-driven clustering approach that operates on historical flight trajectory data and determines well-traveled routes and also less-traveled routes with potential strategic value as reroute options for avoiding Flight Constrained Areas (FCAs). In previous work [9], we used a preliminary version of the clustering approach to generate an Aggregate Route Model (ARM) for strategic planning. We demonstrated that the ARM can accurately predict sector aircraft counts and can be used to pose and solve the strategic planning problem to determine ground delays and pre-departure reroutes for TFM.

Another strategic planner is the Collaborative Trajectory Optimization Program (CTOP) that allows aircraft operators to participate in the decision-making; see, e.g., Refs. [10–12]. For each flight, the operator submits a Trajectory Options Set (TOS) composed of preference-weighted routes from origin to destination. Given predicted FCAs, the options include routes that are more direct but pass through FCAs and could be ground delayed, and routes that avoid FCAs at the expense of increased flight time. CTOP balances the user demand and preferences with the available capacity. The routes produced by the clustering developed in the present paper could serve to populate TOSs.

In [9], the clustering was based on judging the spatial similarity of flight trajectories by the Fréchet distance [13–16]

*As used here, a cell can have various shapes and dimensions depending on the model, e.g., cells can be centers, sectors, or segments of routes.

and the temporal similarity by average ground speed during cruise. The modifications to the clustering process developed in the present paper serve to reduce computation, identify outliers, and automate the selection of the appropriate number of clusters. Given that computing the Fréchet distance between a trajectory pair is costly, an initial coarse-clustering step is introduced to reduce the number of trajectory pairs that must be compared in the subsequent Fréchet distance based fine-clustering step. Flight trajectory undersampling [17, 18] further reduces the Fréchet distance computation time without a significant loss in accuracy. Prior to aggregating the fine-clusters to generate approximate well-traveled routes, outlier trajectories are identified using the methods of [19, 20]. The selection of the number of fine-clusters, and thus the number of aggregate routes, is automated using three performance indices [21–23]. The clustering is controlled by specifying physical distances that are more easily interpreted than the input parameters for the alternative Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [19] clustering used in [8, 24]. In Refs. [24, 25], manual tuning of DBSCAN parameters and visual inspection were used on a case-by-case basis to control the quality of the clusters. For larger numbers of O-D pairs, a more automated approach is desirable.

Other approaches to flight trajectory clustering can be found in [26–28]. In Ref. [26], the Ripley K-function is used to describe the spatial distribution patterns of flight trajectories. Different classes of clusters (multiple, single, dispersed) are identified, and then clustering is performed using DBSCAN. Like other approaches described thus far, outliers are discarded. Refs. [27, 28] use a spectral clustering algorithm with a similarity kernel based on Euclidean distance to identify well-traveled (persistent nominal in their terminology) routes from outliers. An outlier analysis technique using the Fréchet distance, introduced in [29], can distinguish outliers caused by tactical maneuvers (e.g., holding patterns and path stretches) from outliers that may have strategic value.

The paper is organized as follows. Section II establishes the ARM context, describing the basic elements, the characteristics of the historical flight data set, the terminology, and the modeling assumptions. The clustering tools are described in Section III. Section IV introduces the coarse-clustering and describes the procedure for each flight category. Outlier identification is described in Section V. Section VI presents the fine-clustering approach and how the number of fine-clusters for each coarse-cluster is detemined. Section VII explains how a fine-cluster is translated into an aggregate route representative of the cluster. Section VIII describes an application example, and the conclusions are stated in Section IX. The adaptation of the Fréchet distance to paths on a spherical surface is presented in the Appendix.

II. Data-Driven Route Modeling

Our interest is strategic planning for a portion of the NAS, referred to as the planning domain $\mathcal{P}$, and for a time period referred to as the planning horizon. The 3-dimensional planning domain $\mathcal{P}$ is defined as follows. Let A be a connected set on the Earth's surface. A will be taken as the projection of a center, or a union of neighboring centers onto the Earth's surface. The planning domain $\mathcal{P}$ is the airspace above the set A from ground level up to a specified maximum altitude.

The basic elements, i.e., cells, for our Eulerian traffic flow model are one-dimensional segments of aggregate routes; these segments account for the density of aircraft in finite volumes surrounding these segments. To generate the aggregate routes for the domain $\mathcal{P}$ of interest, we 'fine' cluster historical flight data for that domain. In [9], the fine-clusters are computed directly from the historical dataset. In the present paper, to lessen the clustering computational requirements, fine-clustering is preceded by coarse-clustering. And, before computing distances between trajectories for fine-clustering, the data for each trajectory is reduced by a procedure called undersampling. In addition, outlier trajectories are detected; these can be removed or assessed for potential strategic value.

A. Historical Flight Dataset

The selection of the historical dataset on which to base an ARM is an important step; however, it is one we do not address in this paper. The desired dataset would include flights for the aircraft types of interest on clear weather days - flights of sufficient number to extract the well-traveled routes in the domain under consideration. Then for strategic planning on a given day, these clear day routes would be supplemented with additional routes designed specifically to accommodate any forecasted convective weather constraints, or, more generally, any FCAs. The additional routes could come from the Playbook [30] as in current practice, or they could come from the data-driven clustering and aggregation method developed in this paper. In the latter case, strategic routes might be gleaned from the outliers for clear-day data or more likely from well-traveled routes for a time period with FCAs impacting the flow.

Here, the historical dataset is populated from Future ATM Concept Evaluation Tool (FACET) Track (TRX) files [31]. FACET can be run in simulation mode or playback mode. Running original Airport Situation Display to Industry (ASDI) data in playback mode produces TRX files with recorded flight trajectories in a form that is easier to process. For each day (from 00:00 to 23:59 Coordinated Universal Time (UTC)), the TRX file contains information on all the flights in the NAS, with a temporal resolution of 1 minute. The TRX file is divided into 1,440 blocks, one for each minute of the day. For each flight, the information at each time includes aircraft identifier, aircraft type, latitude, longitude, ground speed, heading, altitude, center, sector, flight level, and filed flight plan.

For the purpose of demonstrating our approach, we use a dataset that spans 14 days, from July 1, 2014 to July 14, 2014, which are 'clear days' based on the low WITI (Weather Impacted Traffic Index) for the planning domain comprised of the six western-most centers. The 6 centers are Los Angeles (ZLA), Albuquerque (ZAB), Seattle (ZSE), Oakland (ZOA), Salt Lake City (ZLC), and Denver (ZDV). In $\mathcal{P}$, we only consider airports belonging to the Aviation System Performance Metrics (ASPM) [32] ASPM-77 list. These airports account for the majority of the air traffic flow. The 21 airports considered are listed by center as follows. **ZLA Center**: Burbank Bob Hope (BUR), McCarran International (LAS), Los Angeles International (LAX), Long Beach (LGB), Ontario International (ONT), Oxnard (OXR), Palm Springs International (PSP), San Diego International (SAN), John Wayne (SNA), Van Nuys (VNY). **ZAB Center**: Albuquerque International Sunport (ABQ), Phoenix Sky Harbor International (PHX), Tucson International (TUS).

ZSE Center: Portland International (PDX), Seattle-Tacoma International (SEA). **ZOA Center**: Oakland International (OAK), San Francisco International (SFO), Norman Y. Mineta San José International (SJC), Sacramento International (SMF). **ZLC Center**: Salt Lake City International (SLC). **ZDV Center**: Denver International (DEN).

B. Filtering and Sorting the Dataset

Flight trajectories are divided into subsets by flight category. In [9], the following four flight categories were defined and illustrative examples were given.

- Internal flights: Flights between two airports, with the airports and the entire flight path in the domain $\mathcal{P}$.
- Exiting flights: Flights with origin airport in $\mathcal{P}$, destination airport outside $\mathcal{P}$, and no re-entry into $\mathcal{P}$ after the first exit.
- Entering flights: Flights with destination airport in $\mathcal{P}$ and origin airport outside $\mathcal{P}$.
- Overflights: Flight paths crossing $\mathcal{P}$ with origin and destination airports outside the domain.

We assume that the dataset has a sufficient number of complete trajectories that partial trajectories can be discarded. A partial trajectory is a trajectory in any of the four categories that does not extend, within $\mathcal{P}$, to its origin or destination points. For example, an internal flight is partial if either it was already airborne at the start of the dataset timespan or did not reach its destination by the end of the dataset timespan. Note that for an internal flight, the origin and destination points are the origin and destination airports, respectively. For the other flight categories, the origin and/or the destination point is defined to be the boundary point where the trajectory enters/exits $\mathcal{P}$. For flights in these other categories, trajectory points outside $\mathcal{P}$ are discarded.

Additional filtering of the dataset can be considered to remove trajectories that are unusual. Trajectory clustering, outlier removal, and aggregation will ensure that unusual trajectories do not factor into the resulting ARM, but if an unusual trajectory can be identified and discarded at the outset, it should be to avoid unnecessary computation. The dataset described above provides a specific example. Figure 1 shows the number of flights versus aircraft type, for the dataset described above, ordered from most flights to fewest. Trajectories could be discarded for an aircraft type with too small a sample size, or conversely, if the aircraft type is considered important for strategic planning, the dataset could be expanded. Figure 2 shows, for each aircraft type, the range of average cruise speeds. Note that although the airspeed is the intrinsic characteristic of an aircraft, the TRX files provide ground speed and we are using that. For each aircraft type the mean and standard deviation for the average cruise ground speed are indicated in Fig. 2. For each aircraft type, trajectories with speeds more than one standard deviation from the mean could be discarded. Speed clustering is used to separate aircraft types with significantly different mean speeds.

In the remainder of the paper, we use the terms *flight trajectory*, *flight track* and *ground track*. Flight trajectory refers to the discrete-time sequence of position and velocity obtained from the TRX file. When only the sequence of latitude-longitude-altitude triplets are considered, we call this a flight track. A ground track is the projection of a flight

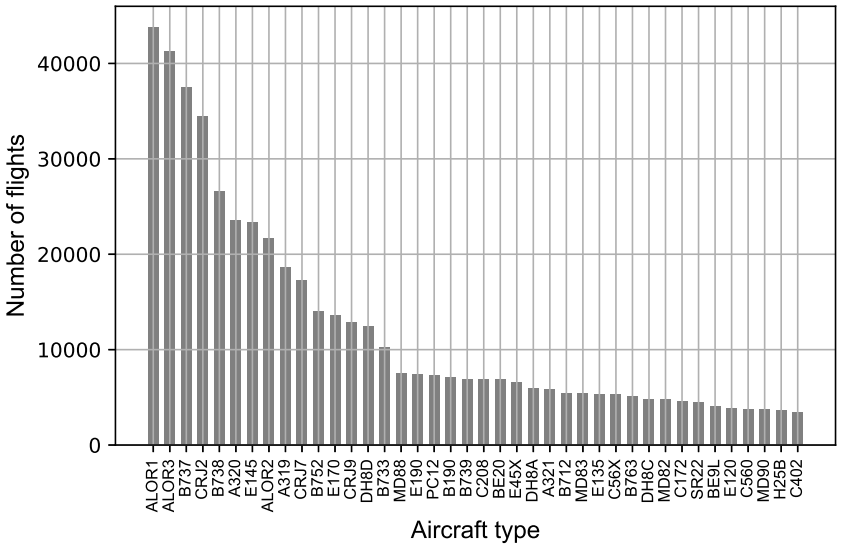


Fig. 1 Number of flights for each aircraft type.

track onto the surface of a sphere; although the surface would normally be earth's surface, when considering the lateral distance between flight tracks in clustering, we translate angular separation into distance at 35,000 feet.

III. Clustering Tools

A. Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering is used to spatially cluster a set of elements based on an appropriate distance metric for assessing the separation between each pair of elements. Initially each element is considered a single cluster. Then a tree structure called a dendrogram is built, which links elements as the maximum intracluster distance is increased. When comparing elements at the lowest level of the dendrogram, the distance between the elements is directly used. To assess the spatial similarity of two clusters, based on the set of distances between pairs of elements, one in each cluster, previous work has used one of three intercluster distances [33]: complete-linkage, single-linkage, or average-linkage. We use the average-linkage, because it yields an appropriate balance between intercluster distance allowed and number of clusters.

Once a dendrogram is created, the number of clusters is determined by selecting a pruning distance. The pruning distance is both the maximum intracluster distance and the minimum intercluster distance. An example of a dendrogram for clustering points is shown in Fig. 3. Thirty points shown in the top left corner are randomly generated inside a unit square, and all pairwise Euclidean distances are computed. Then, using average-linkage distance, the resulting dendrogram is as shown. For a given pruning distance, there is an horizontal line that cuts the dendrogram in as many points as the number of clusters. Pruning distances of 0.01, 0.5 or 0.8 will result in 30, 4 or 1 cluster respectively. Note

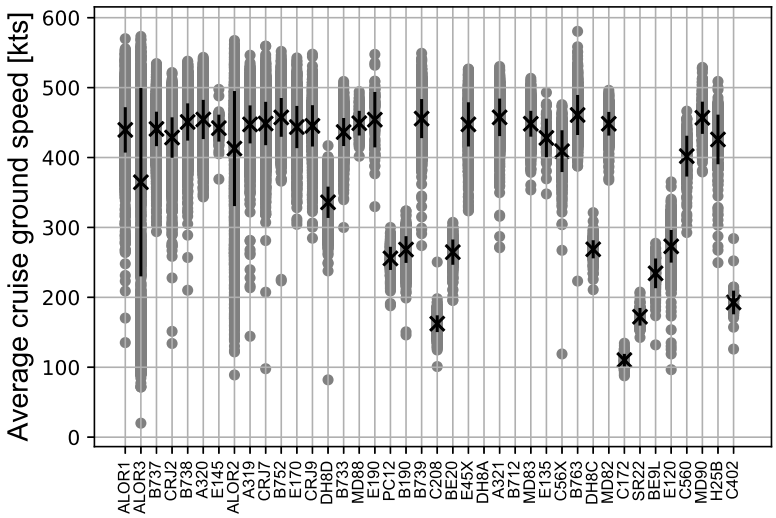


Fig. 2 For each aircraft type, the average ground speed in cruise for each flight (gray circle), the mean speed (cross), and the speeds within one standard deviation of the mean (black line) are given.

that there is always an interval of pruning distances that result in the same number of clusters; for example, any distance between 0.47 and 0.53 results in 4 clusters.

The acceptable range of clustering resolution is established by setting lower and upper bounds on the pruning distance. The lower bound $\underline{d}$ is the least intercluster distance allowed, i.e., the maximum clustering resolution. The upper bound $\overline{d}$ is the greatest intracluster distance allowed, i.e, the minimum clustering resolution. We base the minimum clustering resolution on a characteristic width of the sectors in the planning domain, computed as follows. For each sector, the geometric center is estimated and the line through the center is rotated through 180 deg to find the minimum distance between opposing boundary points. Then the mean of the minimum distances over the sectors in the planning domain is taken to be the characteristic sector width. For fine-clustering, the upper bound $\overline{d}$ is set to half of this width. The objective is avoid clusters whose lateral dimension is large enough that the cluster has substantial portions in two or more sectors, though the approach is not precise in achieving this objective. For fine-clustering, the lower bound $\underline{d}$ is set at 40 km for the maximum clustering resolution. For coarse-clustering, the boundary crossing points are clustered with the pruning distance set at the characteristic sector width, i.e., twice the upper bound for fine-clustering.

The pruning distance for fine-clustering is determined as follows. Given the lower and upper bounds on the pruning distance, there is a feasible set of integer values for the number of clusters N_c. The upper bound $\overline{d}$ dictates the lowest number of clusters N_c^{min}, while the lower bound $\underline{d}$ dictates the highest number of clusters N_c^{max}. Depending on the values of the bounds, there may one or more feasible values of N_c. If there is only one feasible value, then this is the number of clusters. Unless $N_c = 1$ is the only feasible value, it is not considered; for example, if the feasible set is $\{1, 2\}$, we choose $N_c = 2$. If there are two or more feasible values, not including $N_c = 1$, the number of clusters $N_c \in [N_c^{min},$

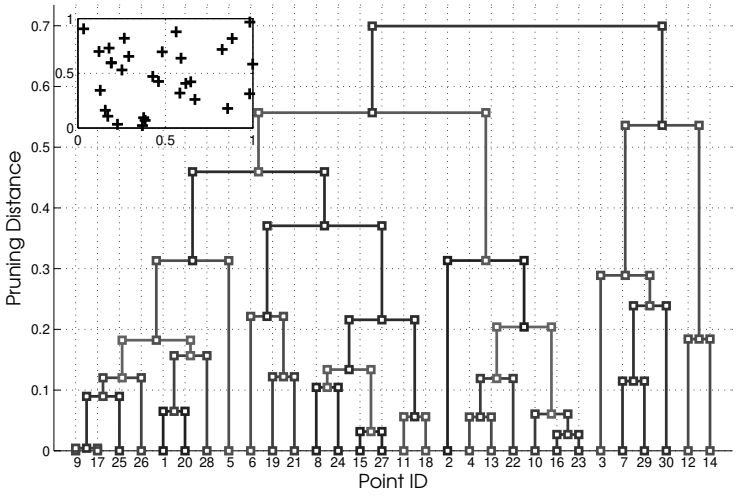


Fig. 3 Example of a dendrogram for a set of points shown in the box in the upper left corner, based on the Euclidean distance between points.

N_c^{max}] is determined using the following three performance indices.

- **Average Silhouette Index** [21]: The silhouette value expresses how similar an element is to its own cluster, compared to other clusters. The silhouette value ranges from -1 to 1, where a high value indicates that the element is well-matched to its own cluster and poorly matched to neighboring clusters. Averaging the silhouette values of all elements, the average silhouette index is obtained.
- **Davies-Bouldin Index** [22]: For each cluster, specific ratios between an intracluster distance and an intercluster distance (see [22] for details) are computed. Then, the average of the different ratios is computed. An index value close to zero indicates an efficient clustering process.
- **Dunn Index** [23]: This index is a ratio between the smallest intercluster distance among all pairs of clusters and the highest intracluster distance. An index value greater than unity indicates an efficient clustering process.

The three performance indices are computed for each of the feasible integer values of N_c. The optimal number of clusters corresponds to the maximum index value for the Average Silhouette and the Dunn indices, and minimum index value for the Davies-Bouldin index. If there is agreement between two or more indices that a particular value of N_c is optimal, this value is selected. If each of the three indices recommends a different number then the median of these is selected.

B. Distance on the Surface of a Sphere

In the coarse-clustering, we need to cluster points on flight tracks, and in the fine-clustering, we need to cluster flight tracks. In both cases, we first project the points or tracks onto the surface of a sphere centered at Earth's center. The rationale is that clustering should focus on the lateral separation of the points and tracks and should account for the

spherical geometry. In specifying distance parameters in the clustering algorithms, we will translate angular separation into the corresponding distance at an altitude of $35,000\ ft$.

For clustering points on the surface of a sphere, the haversine distance is employed. For clustering paths on the surface of a sphere, we use the Fréchet distance [13–16]; readers are referred especially to Ref. [13] for a thorough presentation of this distance metric. The original formulation of the Fréchet distance is for paths in 2D Cartesian space. We have adapted the formulation for paths on the surface of a sphere; see the Appendix for the details. The Fréchet distance, whether in flat or curved space, satisfies the two basic requirements for a distance metric; namely, it is non-negative, and it is zero, if and only if the two paths exactly overlap.

C. Undersampling for Flight Track Data Compression

The cost to compute the Fréchet distance increases quadratically with the number of points with which each flight track is represented [13]. The Ramer-Douglas-Peucker (RDP) algorithm [17, 18] is a means of reducing the number of discrete points representing each track, referred to as undersampling, without compromising the accuracy of the Fréchet distance. Given a distance threshold η, the RDP algorithm undersamples segments of tracks that are quasi-linear with respect to the threshold η, while it retains data points in segments that are not quasi-linear. In the planar case, the algorithm recursively splits sets of points on a path. The splitting point is the point with the maximum Euclidean distance from the line connecting the endpoints of the segment considered. If all the intermediate points of a segment are within a $\pm\eta$ corridor around the segment, only the endpoints are kept.

We adapt the RDP algorithm for flight tracks on a spherical surface, which requires computing distances between points and great circle arcs. Let P_c be a latitude-longitude point, and P_a and P_b the two endpoints of a great circle arc denoted by $\mathcal{A}$. We define the versors $\hat{n}_a$, $\hat{n}_b$ and $\hat{n}_c$ as the vectors from the center of the sphere to points P_a, P_b and P_c, respectively. Then, we compute $\hat{n}_g = \hat{n}_a \times \hat{n}_b$, the versor normal to the plane containing the great circle arc, and $\hat{n}_f = \hat{n}_c \times \hat{n}_g$, the versor normal to a plane that passes through P_c and perpendicular to the plane containing the great circle arc. This latter plane cuts the great circle in two points denote P_1 and $-P_1$. Computing $\hat{n}_{p_1} = \hat{n}_f \times \hat{n}_g$, we obtain the versor from the center of the sphere to one of those two points, while $-\hat{n}_{p_1}$ defines the other point. The last step is to compute the haversine distances η_1=$h(P_c,P_1)$ and η_2=$h(P_c,-P_1)$, where $h(A,B)$ denotes the haversine distance between points A and B on a sphere. The distance between P_c and arc $\mathcal{A}$ is $d(P_c, \mathcal{A})$= min (η_1, η_2). If the two distances η_1 and η_2 are equal, the distance between the point and the great circle arc is $\pi/2$ radians, which is the maximum distance that can be achieved on the sphere between a point and a great circle arc. Figure 4 shows an example where η_1 is the distance between P_c and the great circle arc from P_a to P_b.

Figure 5 depicts the undersampling of two flight tracks from LAX to PHX. We used $\eta = 7.8 \cdot 10^{-5}\ rad$, which corresponds to 500 m on a sphere with of radius $35,000\ ft$. This value provides a good compromise between data reduction and accuracy. The number of data points for projected track P is reduced from 61 to 23, while the number of

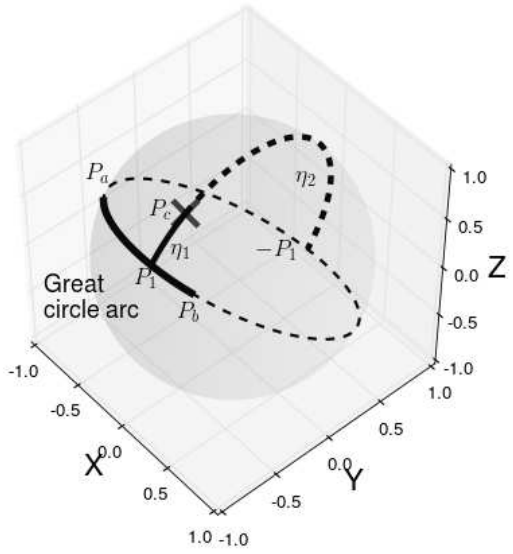
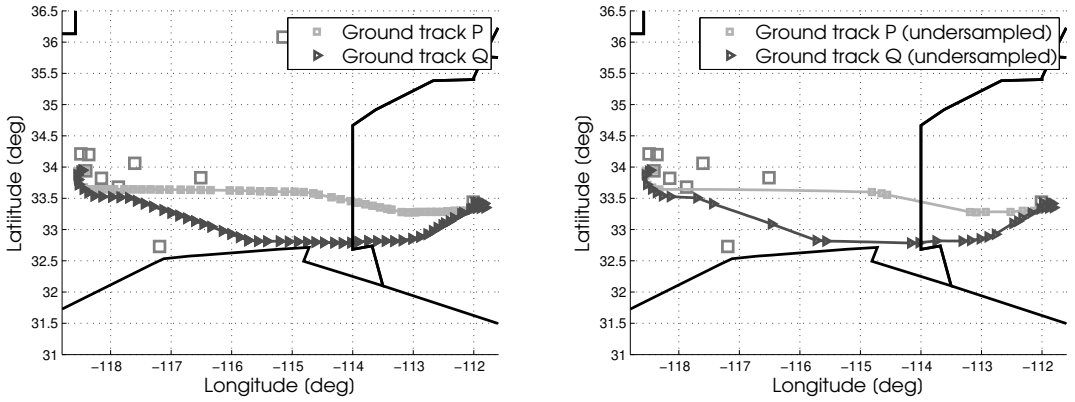


Fig. 4 Graphical interpretation of the distance η_1 between a latitude-longitude pair and a great circle arc.

data points for track Q is reduced from 64 to 32. The Fréchet distance (at $35,000\ ft$) between P and Q is 84.22 km and 84.35 km when using the original and the undersampled tracks, respectively. The error is 0.15%, with the computational time dropping from 4.2 s to 0.5 s. The undersampling procedure typically yields a $50-75\%$ reduction in the number of points. There are larger reductions for longer routes, because the cruise phase, which is quasi-linear, is proportionally larger. On average, the computational time to compute a Fréchet distance is reduced by a factor of 8.



(a) Original ground tracks.

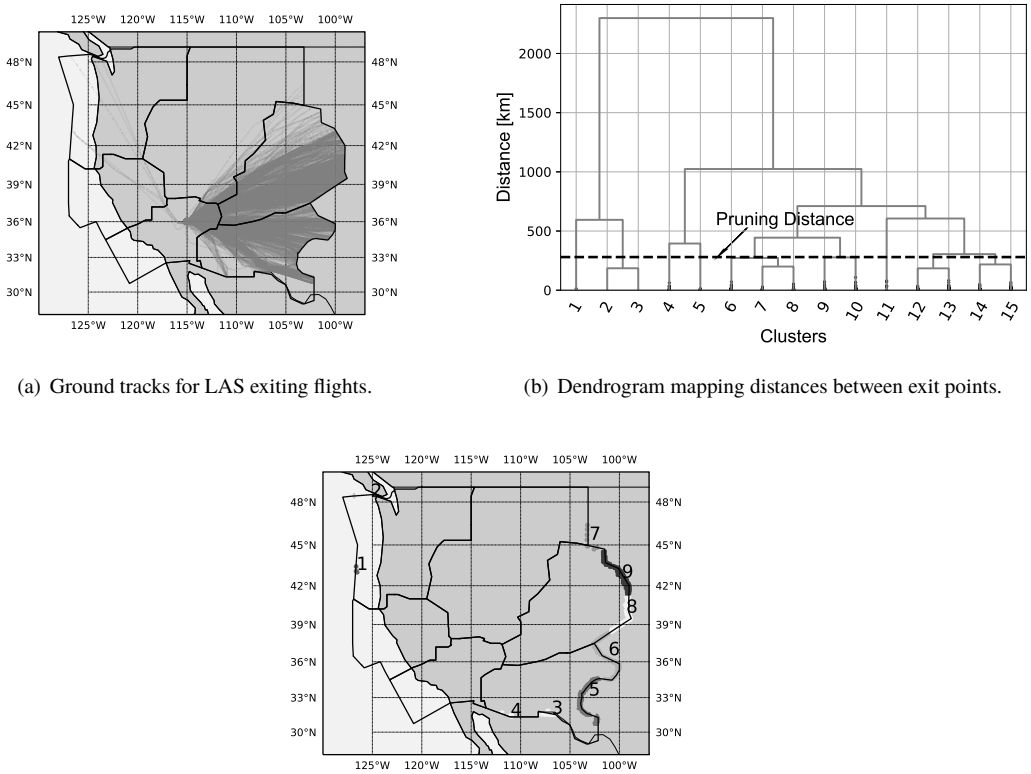
(b) Undersampled ground tracks.

Fig. 5 Two ground tracks from LAX to PHX, before and after undersampling, showing sector boundaries (black) and airports as larger squares.

IV. Coarse-Clustering

Coarse-clustering considers both spatial and speed similarity of flight trajectories. Flight tracks with a common origin and destination are considered spatially similar and grouped together. For internal flights, the origin and destination are airports within the planning domain $\mathcal{P}$. With n_a airports in the domain, there are at most $(n_a - 1)$ destination airports and thus the internal flights for each origin airport are separated into at most $(n_a - 1)$ coarse-clusters.

For an entering-, exiting-, or over-flight, the actual origin and/or destination is outside $\mathcal{P}$. We thus consider the enter/exit point on the boundary of $\mathcal{P}$ to be the origin/destination. For each airport, the coarse-clusters for the exiting flights are determined by clustering the boundary crossing (exiting) points. Similarly, for each airport, the coarse-clusters for the entering flights are determined by clustering the boundary crossing (entering) points. To coarse-cluster the overflights, we first cluster the entry points, and then, for each entry point cluster, we cluster exit points. The boundary crossing point clustering process was described in Sec. III.A.



(a) Ground tracks for LAS exiting flights.

(b) Dendrogram mapping distances between exit points.

(c) Coarse-clusters identified.

Fig. 6 Coarse-clustering of exiting flights from LAS. In subfigures (a) and (c), the boundaries of the 6 western sectors are shown in black.

Coarse-clustering is applied to the outflow from McCarran International Airport (LAS), which serves the Las Vegas area, using the dataset described in Sec. II. Figure 6(a) shows all the ground tracks considered. For this example, the

pruning distance is 280 km, which is the characteristic sector width. Figure 6(b) shows the dendrogram (for visual clarity, only the region of interest is shown). For the given pruning distance, the number of clusters is 9. If Fig. 6(a) shows a few tracks headed northwest, while the majority of the exiting flow is eastward. The boundary crossing points of the two flows define two clusters that are compact and well-separated. However, the intracluster distances are large relative to the characteristic sector width; thus there is further splitting. Figure 6(d) shows the 9 clusters in different shades of gray.

In addition to spatial proximity, coarse-clustering considers average ground speed during the cruise phase, referred to more simply as average cruise speed. The cruise phase trajectory points are determined by discarding the points where the percentile change in altitude between consecutive points is greater than a pre-defined threshold (10% in this paper). In Fig. 7 all the data points within the dashed lines are used in the computation of the average cruise speed. In this particular case, our method identifies the same cruise points as the method [34, 35] which selects all the trajectory points with altitudes above $25,000\ ft$.

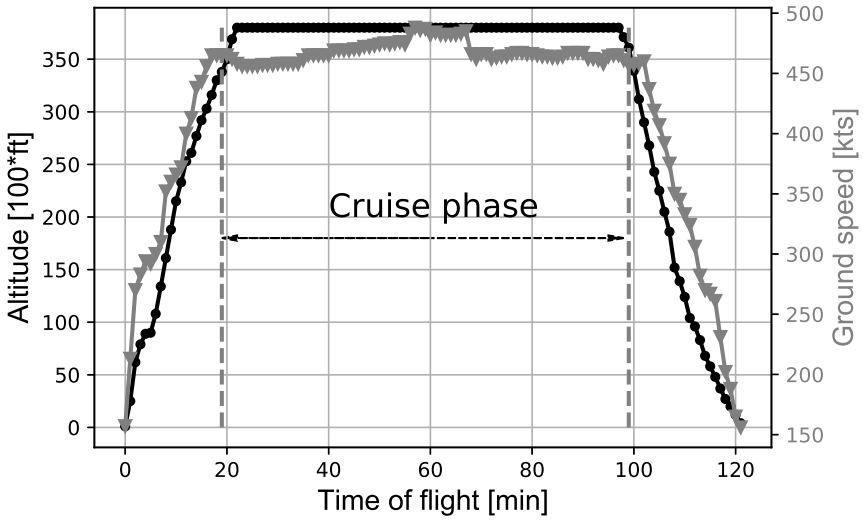


Fig. 7 Cruise phase detection showing altitude points (circles) and the speed points (inverted triangles).

For each coarse-cluster, the average cruise speed distribution is computed. The speed distribution typically has one or two modes. To decide which is the case, the Hartigan dip test [36] is applied. The dip test produces a p-value between 0 and 1. The closer p is to unity the more likely the distribution is unimodal. If the p-value is less than 0.05, there is high confidence that the distribution is bimodal.

A first example of an average cruise speed distribution is shown in Fig. 8 for flight trajectories for the O-D pair Los Angeles International airport (LAX) to Seattle-Tacoma International airport (SEA). The dip test indicates that the distribution is unimodal. The mean is $\mu = 464\ kts$ with a standard deviation of 16.4 kts.

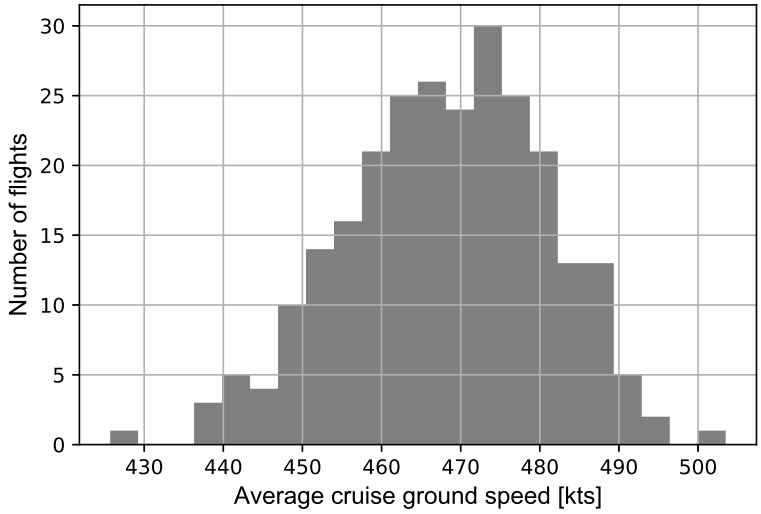


Fig. 8 Average cruise speed distribution for coarse-cluster LAX-SEA.

Figure 9 shows a case where the cruise speed distribution is bimodal. The average cruise speed distribution for flights from ABQ to DEN is plotted. The dip test p-value is 0.008 confirming the bimodal distribution. Thus the trajectories are divided into two clusters. For the first cluster, the mean and standard deviation are $\mu_1 = 343\ kts$ and $\sigma_1 = 6\ kts$. For the second cluster, $\mu_2 = 444\ kts$ and $\sigma_2 = 12\ kts$. Figure 10 shows the correlation between cruise speed and cruise altitude.

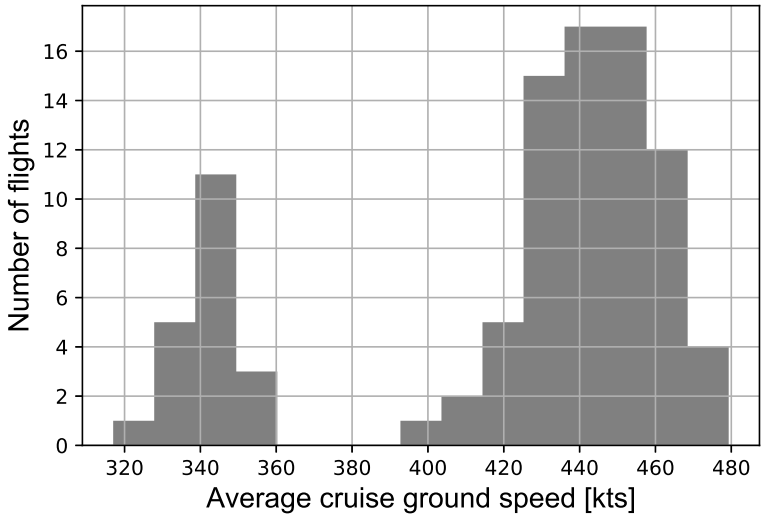


Fig. 9 Average cruise speed distribution for coarse-cluster ABQ-DEN.

To have enough flight trajectories to compute statistical parameters, we discard coarse-clusters with fewer trajectories

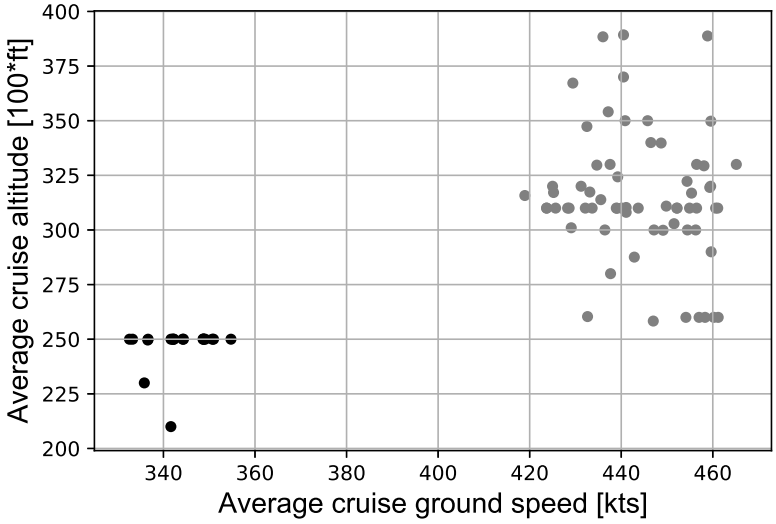


Fig. 10 Average cruise speed versus average cruise altitude for coarse-cluster ABQ-DEN.

than a threshold N_{traj}^{min} (in this work, $N_{traj}^{min} = 15$). Examples of discarded clusters are coarse-clusters 1, 2, and 7 in Fig. 6(c).

V. Outlier Detection

For each coarse-cluster, the trajectories are undersampled, and all pairwise Fréchet distances are computed to assess spatial similarity between ground tracks. The pairwise distances are analyzed for outliers. Two outlier detection methods are used: DBSCAN [19] and the method described in Almeida et al. [20]. Though both methods include outlier detection and clustering, we only use the outlier detection. We use the recommended value for the free parameter controlling each method: for DBSCAN, $MinPts = 5$, and for the Almeida method, $\beta = 1/3$. Any track identified as an outlier by at least one method is discarded.

Fig. 11 shows all the ground tracks from LAX to Salt Lake City International airport (SLC). The outliers are dark gray, and the rest of the tracks are light gray. Out of the 121 original ground tracks, 3 are outliers. Some path stretches and a holding pattern can be seen in the outliers, illustrating that outlier detection and removal serves to eliminate tracks with significant tactical maneuvers.

To construct well-traveled routes, the outliers are deleted from the dataset. We note, however, that the outlier set may contain trajectories that were flown for a strategic purpose. Such trajectories could be considered, for example, in constructing a database (e.g., the Playbook [30]) of reroute options for avoiding FCAs. Routes could be selected from such a database to populate a TOS for CTOP or to augment other route-based traffic models such as the ARM. Figures 12 (a), (b) and (c) show three different outliers associated with traffic flow from LAX to, respectively, SLC, PDX and Philadelphia International airport (PHL). In each figure, the outlier is depicted in dark gray, its nearest neighbor (the

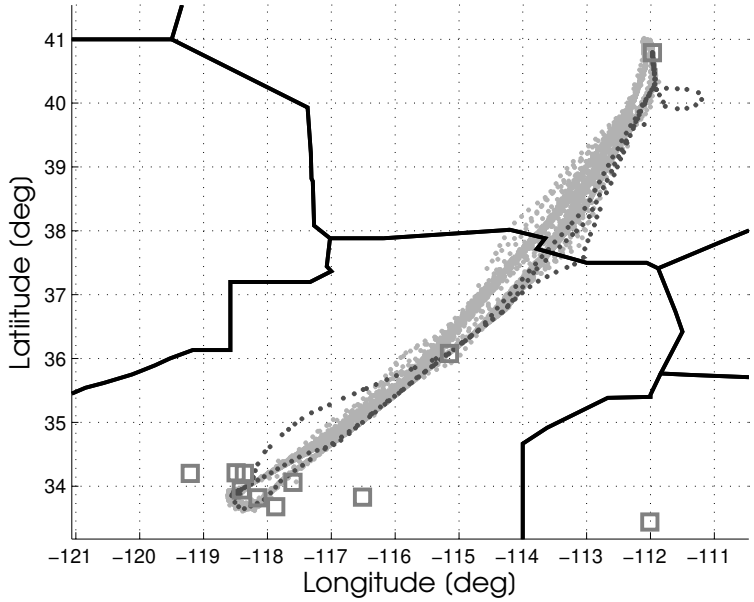
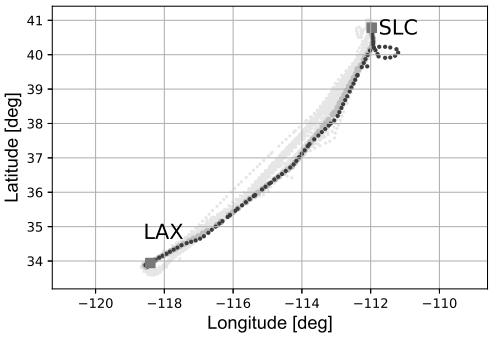


Fig. 11 Projected tracks of the LAX-SLC coarse-cluster showing outliers (dark gray), the rest of the tracks (light gray), sector boundaries (black), and airports (gray squares).

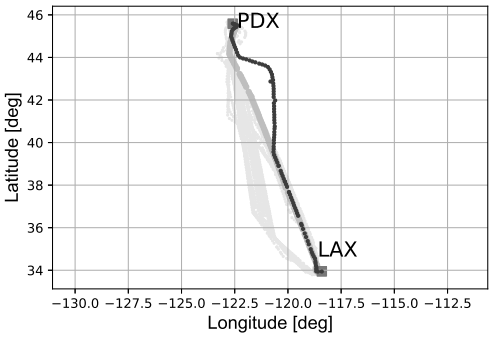
non-outlier ground track with the smallest Fréchet distance from the outlier) in gray, and the other non-outlier ground tracks in light gray. The outliers in Figs. 12(a) and (b) have tactical maneuvers: a holding pattern and a path stretch, respectively. On the other hand, the outlier shown in Fig. 12(c) appears to be a pre-departure reroute for a strategic purpose. The method presented in the appendix of Ref. [29] is a means of sorting outliers into tactical and potentially strategic subsets; the Fréchet distance is used to compute an overlap index between two flight tracks, where the overlap index is the fraction of the length of the nearest neighbor over which the outlier track is within a specified distance of the nearest neighbor.

VI. Fine-Clustering

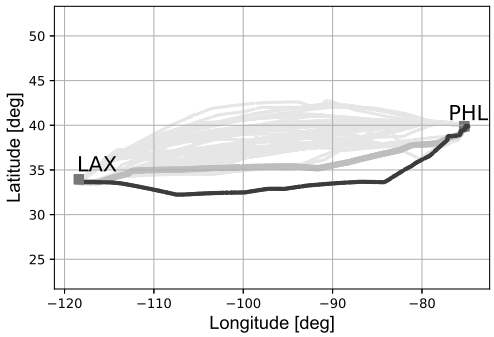
Each outlier-free-coarse-cluster is fine-clustered using the Fréchet distance and hierarchical clustering described in Sec. III. Figure 13 shows an example of the process for the LAX-SEA O-D airport pair. The coarse-cluster had 185 ground tracks. The outlier detection algorithm identified 22 outliers. The outliers and the remaining 163 tracks are shown in dark and light gray, respectively, in Fig. 13(a). Figure 13(b) shows the dendrogram computed using the 13,203 pairwise Fréchet distances. Based on the characteristic sector width, the upper bound $\overline{d}$ is set at 140 km, thus the minimum allowed number of clusters is 2. The 40 km lower bound cuts the dendrogram in 6 different points. Thus the range for N_c is 2-6. Figure 13(c) shows the values of the performance indices. Recalling that the Dunn and Average Silhouette are maximized and the Davis-Boldin index is minimized, all three indices indicate that the number of clusters should be 2. The resulting two fine-clusters are shown in Fig. 13(d).



(a) Outlier due to holding pattern.



(b) Outlier due to path stretch.



(c) Strategic outlier.

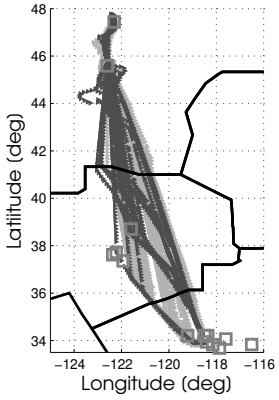
Fig. 12 Different outlier types.

VII. Aggregate Route Computation

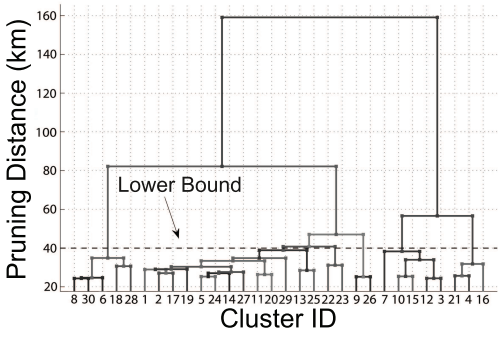
Each fine-cluster constitutes a well-traveled path in the NAS that has been flown by aircraft with similar average cruise speeds. Each fine-cluster is represented by an aggregate route. An aggregate route is an ordered sequence of points in 3D position space; each point is the centroid of the points, one per fine-cluster trajectory, with the same elapsed time since departure (from an origin airport or when entering $\mathcal{P}$). Prior to computing the centroids, we discard some trajectories and resample others so that the trajectories being aggregated are represented by the same number of discrete points and the points at the same elapsed time along the respective trajectories are closer together.

For each fine-cluster, the mean and standard deviation for the distribution of times of flight are computed. Trajectories whose time of flight differs from the mean by more than one standard deviation are discarded. The remaining trajectories are more uniform, but may still have unequal numbers of data points. For trajectories with shorter times of flight, resampling is used to add points where the cruise speed is highest. Similarly, for trajectories with longer times of flight, resampling is used to remove points where the cruise speed is lowest.

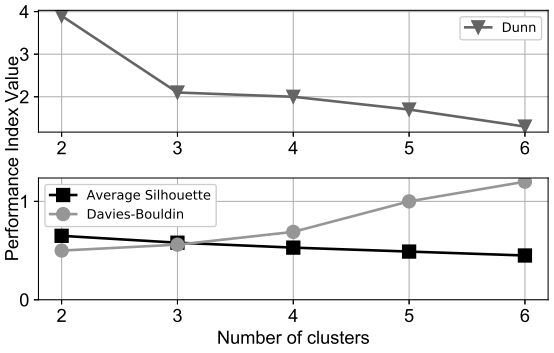
At each discrete time, the centroid of the points on the trajectories of the fine-cluster is computed as a triplet of latitude, longitude, and altitude. The time-ordered sequence of centroids is the aggregate route. For each point of the



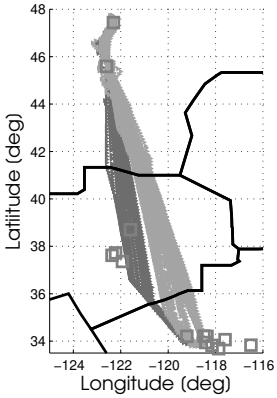
(a) Ground tracks, with outliers in dark gray.



(b) Dendrogram used in the clustering process.



(c) Performance indices.



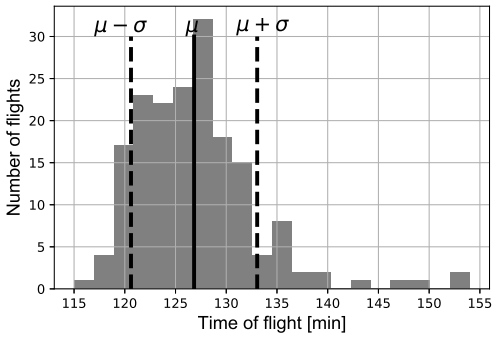
(d) The two fine-clusters in different shades of gray.

Fig. 13 Fine-clusters for the LAX - SEA O-D pair. Subfigures (a) and (c) show sector boundaries (black) and airports (squares).

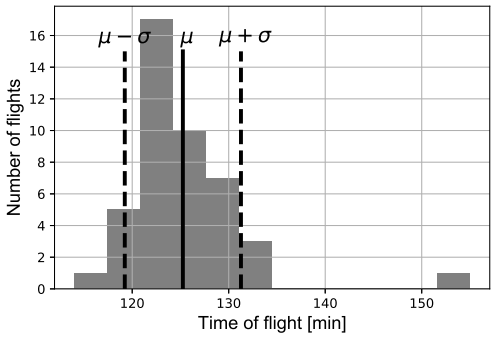
aggregate route, the cruise speed is the mean cruise speed. The flight level, center and sector for each point is assigned based on centroid position.

Figure 14 shows an example of how fine-clusters are translated into aggregate routes. Flight trajectories for the same airport pair described in Sec. II are considered. Figures 14(a) and 14(b) show the time of flight distributions for the left and right fine-clusters of Fig. 13(d). The left fine-cluster has an aggregate time of flight of 126 minutes, compared to the 127 minutes of the right fine-cluster. For the left fine-cluster, flight trajectories with times of flight between 119 and 131 minutes are used to compute the aggregate route (20 out of 33 tracks). For the right fine-cluster, flight trajectories with times of flight between 121 and 133 minutes are used to compute the aggregate route (90 out of 130 tracks). The two resulting aggregate routes are shown in Fig. 14(c). Figure 15(a) shows the distribution of the latitude-longitude pairs and the resulting centroids (for visual clarity, only every tenth centroid is shown) that form the aggregate route, for the right fine-cluster. Figure 15(b) is a zoomed view of the fine-cluster points represented for each aggregate route point.

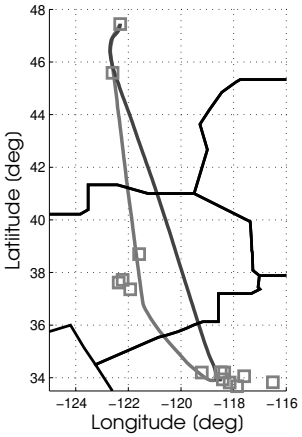
The aggregate routes are not constrained by specified waypoints or fixes. Given that most filed flight plans comply



(a) Time of flight distribution for the left fine-cluster.



(b) Time of flight distribution for the right fine-cluster.

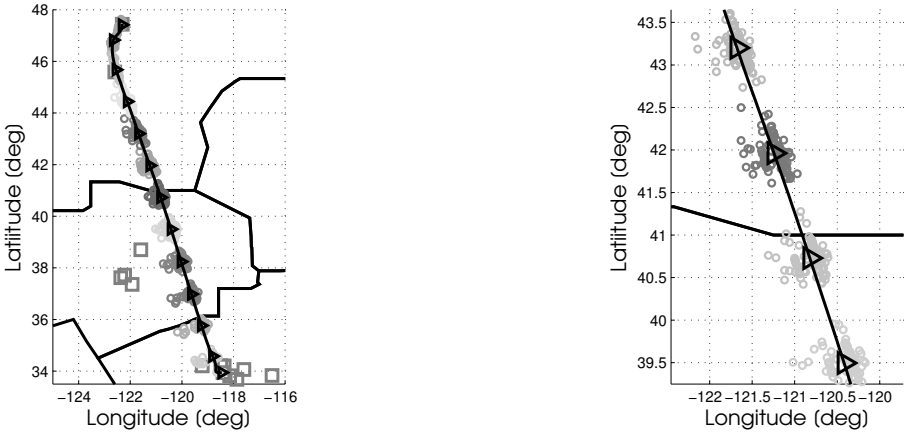


(c) Aggregate routes, sector boundaries (black), and airports (squares).

Fig. 14 Time of flight distributions, with mean σ and standard deviation σ, and the resulting aggregate routes for the LAX-SEA coarse-cluster.

with Standard Instrumental Departure (SID) for take-off, waypoints for the enroute phase, and Standard Terminal Arrival Routes (STAR) for landing, and that for most flights these plans are followed, it is not surprising that our approximations of well-traveled routes usually conform to these fixes and waypoints. However, generating aggregate routes from trajectory data may also identify routes not associated with usual operations in the enroute and terminal area phases. As mentioned earlier, some of these routes may have strategic value as reroutes in response to FCAs or they may prove to be usual routes and suggest the need for addtional waypoints and fixes [27].

An aggregate route has a graph theoretic interpretation. An aggregate route is an ordered sequence of nodes. Aircraft enter the route at the origin (first) node, move along the route, and exit at the destination (last) node. Each aggregate route is a directed graph $\mathcal{G}_r = (\mathcal{N}_r, \mathcal{E}_r)$, where $\mathcal{N}_r$ is the set of nodes forming the aggregate route with n_n the number of nodes, while $\mathcal{E}_r$ is the set of $(n_n - 1)$ edges. The union of all directed graphs defines the full graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ that approximates the air traffic flow inside $\mathcal{P}$. A thorough description of the graph structure of the ARM can be found in [37]. The graph can be used in an optimization framework, where TFM controls such as ground holding and



(a) Full route.

(b) Zoomed view.

Fig. 15 For the LAX-SEA right fine-cluster, ground tracks points (circles), centroids (triangles), aggregate route, sector boundaries (black), and airports (squares).

pre-departure rerouting can be defined using the properties of the graph [9].

VIII. Application to the Six Western-Most Centers of the NAS

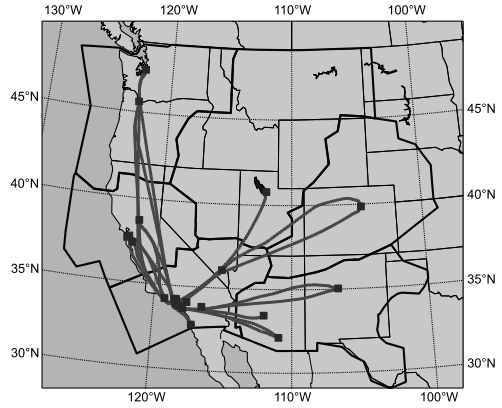
We now demonstrate clustering and aggregation for the six western-most centers using the historical dataset described in Sec. II covering 14 days, July 1, 2014 to July 14, 2014. For each of the 19 airports considered, internal, exiting, and entering flight trajectories are processed. Note that, although there are 21 airports, we do not include the general aviation airports OXR and VNY. The same clustering parameters given in the previous sections are used.

First the flights are coarse-clustered. Regarding internal flights, for 19 airports there are 234 different O-D airport pairs; however, only 126 of these have significant air traffic flow. Eleven of the O-D flows are found to have bimodal cruise speed distributions; each of these is split into two coarse-clusters. Thus the internal flows are split into 137 coarse-clusters. The 19 entering flows are separated into 61 coarse-clusters. The 19 exiting flows are separated into 72 coarse-clusters. The number of Fréchet distance computations is reduced from 14,363,105 to 7,258,133 by coarse-clustering the entering flows, and from 10,789,565 to 4,971,956 by coarse-clustering the exiting flows.

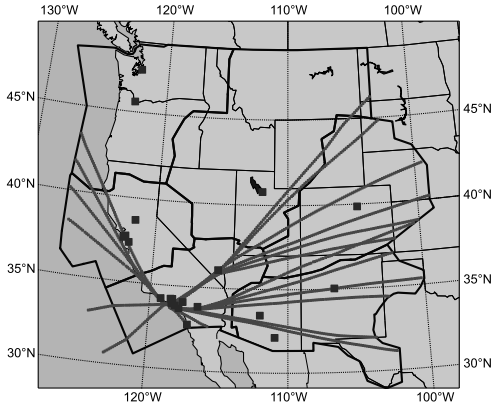
The next steps are outlier detection, fine-clustering, and aggregation. The results are summarized in Table 1. For all centers except ZLA, the algorithm identified internal, exiting and entering flows for each airport. In ZLA, the primarily regional airports BUR and PSP have only internal aggregate routes. BUR and PSP are regional airports. For internal flights, a total of 17,400 trajectories are processed and, after outlier removal, 15,713 trajectories are aggregated into 364 routes. For exiting flights, 13,744 trajectories are processed and, after outlier removal, 12,641 trajectories are aggregated into 156 routes. For entering flights, 15,854 trajectories are processed and, after outlier removal, 14,466 trajectories are aggregated into 184 routes. Figure 16 shows the internal, exiting, and entering aggregate routes for LAX.

Table 1 Number of trajectories before (IGT) and after (GT) outlier removal and number of aggregate routes (AR), for each airport in the planning domain.

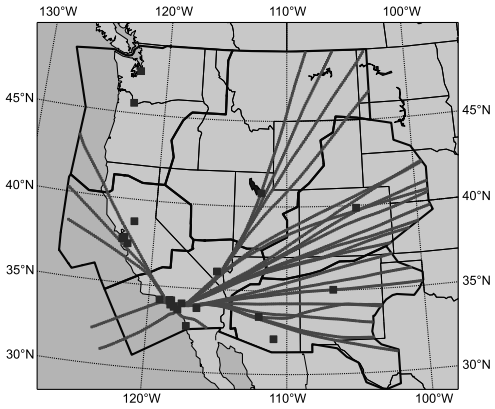
	Internal			Exiting			Entering		
	IGT	GT	AR	IGT	GT	AR	IGT	GT	AR
BUR	423	376	11	38	12	×	38	14	×
LAS	1613	1468	33	1695	1515	12	1763	1612	21
LAX	2083	1930	21	2782	2627	20	3375	3197	27
LGB	244	207	10	77	65	3	66	33	1
ONT	379	338	13	98	87	5	136	82	4
PSP	118	97	10	10	×	×	12	7	×
SAN	1089	1005	20	640	595	10	809	661	14
SNA	601	516	17	240	202	8	319	236	8
ABQ	318	252	12	365	343	5	391	364	4
PHX	1728	1549	27	1424	1305	15	1892	1757	9
TUS	187	135	8	137	121	5	166	139	4
PDX	1064	973	26	436	384	9	517	424	13
SEA	1728	1565	25	998	916	8	988	858	18
OAK	861	801	17	157	119	4	223	169	6
SFO	1490	1382	22	1507	1423	18	2258	2173	22
SJC	599	540	14	130	120	4	202	182	6
SMF	547	509	18	135	122	6	195	180	8
SLC	1186	1073	33	791	729	14	841	796	13
DEN	1142	997	27	2084	1956	10	1663	1582	6
Total	17400	15713	364	13744	12641	156	15854	14466	184



(a) Internal aggregate routes.



(b) Exiting aggregate routes.



(c) Entering aggregate routes.

Fig. 16 Internal, exiting, entering aggregate routes for LAX airport with sector boundaries (black) and airports (squares).

IX. Conclusions

An approach for identifying and approximating well-traveled routes in a historical dataset of flight trajectories has been presented. The routes are produced by clustering and aggregation; the clustering considers both spatial and speed similarity. Fine-clusters are aggregated, following outlier detection and removal, into routes. The fine-clustering is based on the Fréchet distance metric, adapted for paths on a spherical surface. The computational requirements for clustering and aggregation are reduced by an initial coarse-clustering step that considers origin, destination, and average cruise speed, and by undersampling. The selection of the number of clusters is automated after setting physical parameters that bound the intra- and inter-intercluster distances. The effectiveness of the clustering approach was demonstrated on a historical dataset for a planning domain composed of 6 centers with a total of 19 airports. The routes produced by clustering and aggregation can used in strategic planning.

Appendix - Fréchet Distance Computation for Great Circle Arc Splines

The original formulation for the Fréchet distance computation is modified to account for the different geometry of the problem. The original formulation [13] computes distances between trajectories that lie on a 2D plane. In our case, we compute distances between tracks that are projected onto the surface of a sphere, which requires a reformulation. The original framework is first briefly discussed to provide the background for the reformulation.

Reference [13] describes into detail how the Fréchet distance $\mathcal{F}$ between two polygonal curves (i.e., piecewise-linear planar curves) is computed. A first polygonal curve P with p edges (and $p + 1$ data points), and a second polygonal curve Q with q edges (and $q + 1$ data points) are given. Then, a matrix with q rows of p horizontally stacked unitary squares is created. The unitary square in position (i, j) maps distances (generally Euclidean in planar applications) between the i-th normalized edge of Q and the j-th normalized edge of P. Given a candidate distance $\mathcal{F}_\epsilon$, for each unitary square it is computed the locus of points whose distance is less or equal to $\mathcal{F}_\epsilon$. This locus is called free space [13], and is convex. The process is repeated for all unitary squares, and the free space diagram for the two polygonal curves is obtained. Note that, while each free space is convex, the union of all free spaces (i.e., the free space diagram) is not necessarily convex. In the free space diagram, the computation of $\mathcal{F}$ is subject to the solution of a decision process. A monotone path from the upper left corner (first data point of both curves) to the lower right corner (last data point of both curves) is searched, since curves need to be entirely spanned [13]. Due to a monotonicity requirement that arises from a backtracking constraint, we are only interested in the intersections of each free space with the four sides of the associated unitary square. The four intersections are defined $L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$ in [13]. The free space diagram can be interpreted as a directed graph. A different candidate distance $\mathcal{F}_\epsilon$ will result in a different set of $L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$ ($1 \leq i \leq q$, $1 \leq j \leq p$), and in different adjacency properties of the directed graph. The smallest distance such that a monotone path compatible with the adjacency properties is found, is the Fréchet distance $\mathcal{F}$. Figure 17 shows an example of free space diagram with $p = 33$ and $q = 18$, which is the union of 594 free spaces. A monotone path from the upper left corner to the lower right corner within the gray region is identifiable. Given our reference choice for the free space diagram, a monotone path is a path that develops downwards or towards the right.

In most applications, the Fréchet distance is applied to polygonal curves. $L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$ are computed as intersections of circles with segments, i.e., as solutions of a quadratic equation. For the purpose of calculating the Fréchet distance in our application, consecutive ground track latitude-longitude pairs are connected by a great circle arc to obtain a continuous curve. The different framework implies a different approach in the computation of $L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$. Polygonal curves in the Cartesian formulation become great circle arc sequences. Likewise, the Euclidean distance is replaced with the haversine distance. The solution of the decision process follows the same procedure mentioned above instead.

Each edge of a free space, as the one shown in Fig. 17, maps the mutual distance between a great circle arc of a track, and a latitude-longitude pair of the other track. For each unitary square, determining the existence and extension

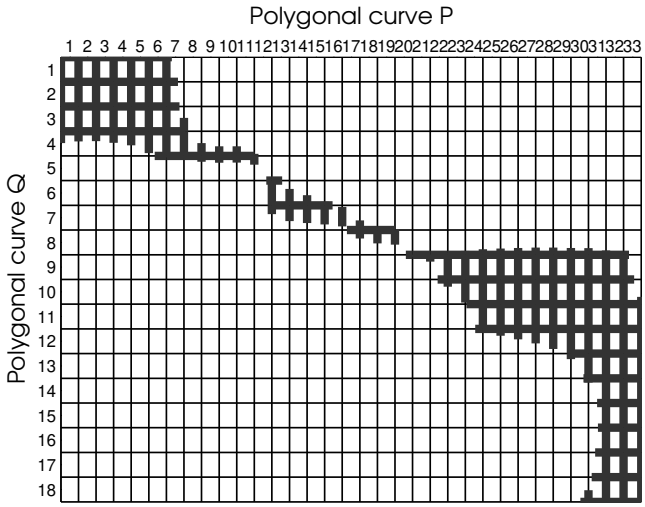
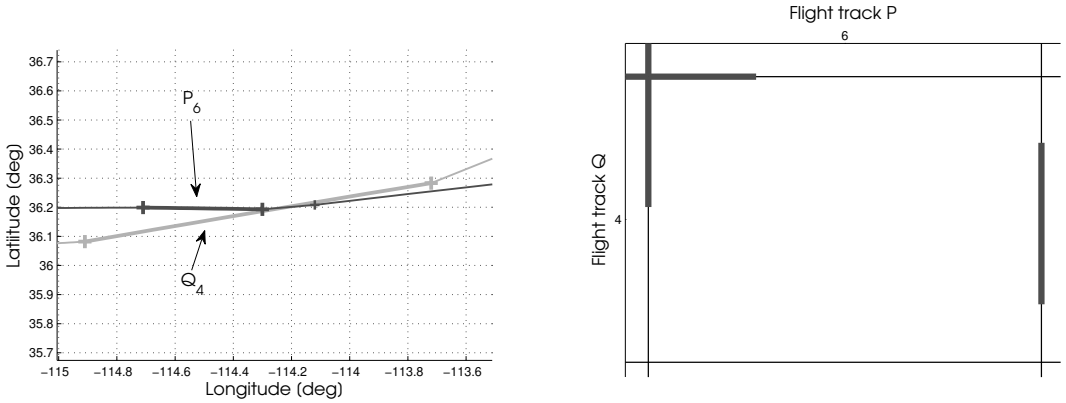


Fig. 17 Example of a free space diagram.

of $L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$ involves two latitude-longitude pairs per track, and one great circle arc per track connecting the latitude-longitude pairs.

$L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$ are computed detecting a possible intersection between a circle centered in a latitude-longitude pair of curve P and a great circle arc of curve Q (or vice versa). The radius of the circle is the candidate distance $\mathcal{F}_m$. Because each square is unitary, each intersection ranges between 0 and 1 (meaning respectively that the tail/head node of the great circle arc is inside the circle).



(a) Two great circle arcs, belonging to different ground tracks.

(b) Associated unitary square with $\mathcal{F}_m = 30.5\ km$.

Fig. 18 Example of how $L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$ are computed for a unitary square. In subfigure (a), the 4 tail/head latitude-longitude pairs are indicated with plus signs and the 2 great circle arcs are in bold.

Figure 18 shows a visual example of the four intersections $L^F_{i,j}$, $L^F_{i,j+1}$, $B^F_{i,j}$, $B^F_{i+1,j}$. On the left, the sixth great circle arc P_6 of ground track P and the fourth great circle arc Q_4 of ground track Q are highlighted in gray. On the right, the

associated unitary square is depicted. For the upper edge $B^F_{4,6}$, there is an intersection ranging from 0 to roughly 0.25. Thus, a circle with radius $\mathcal{F}_m$, centered in the left vertex of Q_4, contains the first quarter of P_6. For the right edge $L^F_{4,7}$, the intersection spans roughly from 0.25 to 0.75. Thus, the same circle, centered in the right vertex of P_6, contains the central part of Q_4. No intersection is identified for the lower edge, which means that a circle with radius $\mathcal{F}_m$ centered in the right vertex of Q_4 does not contain any portion of P_6.

In the spherical formulation, the intersection of circles with great circle arcs is computed as solution of a trigonometric equation. Given a point on a sphere of radius r_s defined with the triplet latitude-longitude-radius, the equivalent position in Cartesian coordinates is $\hat{x} = [x\ y\ z]$. The position of the center of the circle (i.e., a vertex of a great circle arc) is defined by the vector $\hat{x}_c$. The position of the endpoints of a great circle arc belonging to the other curve is defined by vectors $\hat{x}_1$ and $\hat{x}_2$ respectively. The length of the great circle arc is d. The goal is to determine if the great circle containing the great circle arc between $\hat{x}_1$ and $\hat{x}_2$ intersects the locus of points that (i) belong to the sphere and (ii) have a haversine distance of $\mathcal{F}_m$ with respect to $\hat{x}_c$. The great circle arc coordinates $\hat{x}$ are parametrized with parameter f ($f \in [0, 1]$), such that $\hat{x} = \hat{x}_1$ if $f = 0$ and $\hat{x} = \hat{x}_2$ if $f = 1$. Using spherical trigonometry properties, we can write

$$\hat{x} = \frac{\sin((1-f)d)}{\sin(d)}\hat{x}_1 + \frac{\sin(fd)}{\sin(d)}\hat{x}_2 \tag{1}$$

while requirements (i) and (ii) are translated into the following set of equations

$$x^2 + y^2 + z^2 = r_s^2 \tag{2}$$

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = 4r_s^2 \sin^2\left(\frac{\mathcal{F}_m}{2}\right) \tag{3}$$

Substituting Eq. 2 into Eq. 3, the following locus is obtained

$$xx_c + yy_c + zz_c = r_s^2\left[1 - \sin^2\left(\frac{\mathcal{F}_m}{2}\right)\right] \tag{4}$$

and combined with Eq. 1. Expanding $\sin((1-f)\,d)$ using trigonometric identities, the final trigonometric equation is written in compact form as

$$C_1\cos(fd) + C_2\sin(fd) = C_3 \tag{5}$$

where

$$\begin{cases} C_1 = x_1 x_c + y_1 y_c + z_1 z_c \\ C_2 = \dfrac{x_2 x_c + y_2 y_c + z_2 z_c - \cos(d)\,(x_1 x_c + y_1 y_c + z_1 z_c)}{\sin(d)} \\ C_3 = r_s^2 \left[1 - \sin^2\left(\dfrac{\mathcal{F}_m}{2}\right)\right] \end{cases}$$

Equation 5 can be rewritten as

$$R\cos(f\,d - \gamma) = C_3, \quad \text{with} \quad R = \sqrt{C_1^2 + C_2^2}, \quad \gamma = \tan^{-1}\left(\frac{C_2}{C_1}\right) \tag{6}$$

For a free space diagram, the overall number of $L_{i,j}^F$ to be computed is $(p+1)\,q$. The overall number of $B_{i,j}^F$ is $(q+1)\,p$. Thus, Eq. 6 needs to be solved $2pq + q + p$ times. The existence of feasible values for $L_{i,j}^F$ and $B_{i,j}^F$ depends on $\cos^{-1}\left(\dfrac{C_3}{R}\right)$ as follows

1) $\cos^{-1}\left(\dfrac{C_3}{R}\right) > 1$: the great circle containing the great circle arc from $\hat{x}_1$ to $\hat{x}_2$ does not intersect the locus defined by Eq. 4. No feasible interval for f is obtained.
2) $\cos^{-1}\left(\dfrac{C_3}{R}\right) = 1$: the great circle containing the great circle arc from $\hat{x}_1$ to $\hat{x}_2$ is tangent to the locus defined by Eq. 4. A single solution $f_1 = f_2 = \bar{f}$ is obtained. If $0 \leq \bar{f} \leq 1$, the solution is feasible, otherwise the point of tangency is external with respect to the great circle arc.
3) $\cos^{-1}\left(\dfrac{C_3}{R}\right) < 1$: the great circle containing the great circle arc from $\hat{x}_1$ to $\hat{x}_2$ intersects the locus defined by Eq. 4 in two distinct points. Two solutions f_1 and f_2 are obtained. If
    1) $0 \leq f_1 \leq 1$ and $0 \leq f_2 \leq 1$: both solutions are feasible, and the interval of validity for the parameter is $f = [\min(f_1, f_2) \quad \max(f_1, f_2)]$.
    2) $0 \leq f_1 \leq 1$ and $f_2 > 1$ or vice versa: one solution is feasible. Define $\bar{f} = \min(f_1, f_2)$ and d_1 and d_2 the haversine distances between $\hat{x}_c$ and $\hat{x}_1$, and $\hat{x}_c$ and $\hat{x}_2$ respectively. If $d_1 < d$, the valid interval for the parameter $f = \left[0 \quad \bar{f}\right]$, otherwise $f = \left[\bar{f} \quad 1\right]$.
    3) $f_1 > 1$ and $f_2 > 1$: the great circle arc from $\hat{x}_1$ to $\hat{x}_2$ is either completely contained in the locus, or totally external. If $d_1 > d$ and $d_2 > d$, no valid interval for the parameter is obtained. If $d_1 < d$ and $d_2 < d$, $f = [0 \quad 1]$.

The $2pq + q + p$ intervals computed solving Eq. 6, as well as the monotonicity requirement, determine the adjacency properties of the directed graph mapping the free space diagram. A depth-first-search solver [38, 39] is then used to determine the smallest distance value such that, in the associated free space diagram, a monotone path between node 1 (upper left corner) and node $3pq + 2\,(p+q) + 1$ (lower right corner) exists. That value identifies the Fréchet distance $\mathcal{F}$.

Funding Sources

The material is based upon work partially supported by the National Aeronautics and Space Administration under Prime Contract Number NAS2-03144 awarded to the University of California, Santa Cruz, University Affiliated Research Center.

References

[1] FAA Website, "https://www.faa.gov/nextgen/programs/weather/tfm_support/overview/", 2018, Accessed: 2018-June-20

[2] Bertsimas, D., and Patterson S. S., "The Air Traffic Flow Management Problem with Enroute Capacities", *Operations Research*, Vol. 46, pp. 406–422, 1998

[3] Bertsimas, D., Lulli, G., and Odoni, A., "An Integer Optimization Approach to Large-Scale Air Traffic Flow Management", *Operations Research*, Vol. 59, No. 1, pp. 211–227, 2011, doi:10.1287/opre.1100.0899

[4] Bayen, A. M., Grieder, P., Meyer, G., and Tomlin, C. J., "Lagrangian Delay Predictive Model for Sector-Based Air Traffic Flow", *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, pp. 1015–1026, 2005

[5] Menon, P. K., Sweriduk, G. D., and Bilimoria, K. D., "New Approach for Modeling, Analysis, and Control of Air Traffic Flow", *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, pp. 737–744, 2004, doi:10.2514/1.2556

[6] Sridhar, B., Grabbe, S. R., and Mukherjee, A., "Modeling and Optimization in Traffic Flow Management", *Proceedings of the IEEE*, Vol. 96, No. 12, pp. 2060–2080, 2008, doi:10.1109/JPROC.2008.2006141

[7] Sun, D., and Bayen, A. M., "Multicommodity Eulerian-Lagrangian Large-Capacity Cell Transmission Model for En Route Traffic", *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, pp. 616–628, 2008, doi:10.2514/1.31717

[8] Marzuoli, A., Gariel, M., Vela, P. A., and Feron, E., "Data-Based Modeling and Optimization of En Route Traffic", *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, pp. 1930–1945, 2014, doi:10.2514/1.G000010

[9] Bombelli, A., Soler, L., Trumbauer, E., and Mease, K. D., "Strategic Air Traffic Planning with Fréchet Distance Aggregation and Rerouting", *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 5, pp. 1117–1129, 2017, doi:10.2514/1.G002308

[10] Arneson, H., Evans, A. D., Kulkarni, D., Lee, P., Li, J., and Wei, M. Y., "Using an Automated Air Traffic Simulation Capability for a Parametric Study in Traffic Flow Management", *AIAA Aviation Technology, Integration, and Operations Conference*, 2018, doi:10.2514/6.2018-3665

[11] Murça M. C. R., "Collaborative air traffic flow management: Incorporating airline preferences in rerouting decisions", *Journal of Air Transport Management*, Vol.71, pp. 97–107, 2018, doi:10.1016/j.jairtraman.2018.06.009

[12] Rodionova, O., Arneson, H., Sridhar, B., Evans, A. D., "Efficient trajectory options allocation for the collaborative trajectory options program", *Digital Avionics Systems Conference*, 2017

[13] Alt, H., and Godau, M., “Computing the Fréchet Distance Between Two Polygonal Curves", *International Journal of Computational Geometry & Applications*, Vol. 5, No. 01n02, pp. 75–91, 1995, doi:10.1142/S0218195995000064

[14] Alt, H., Knauer, C., and Wenk, C., “Matching Polygonal Curves with Respect to the Fréchet Distance", *18th Annual Symposium on Theoretical Aspects of Computer Science*, pp. 63–74, 2001

[15] Chen, D., Driemel, A., Guibas, L. J., Nguyen, A., and Wenk, C., “Approximate Map Matching with Respect to the Fréchet Distance", *Proceedings of the 13th Workshop on Algorithm Engineering & Experiments*, pp. 75–83, Society for industrial and Applied Mathematics, 2011

[16] Maheshwari, A., and Yi, J., “On Computing Frèchet Distance of Two Paths on a Convex Polyhedron", *Proceedings of the 21st European Workshop on Computational Geometry*, pp. 41–44, 2005

[17] Ramer, U., “An iterative procedure for the polygonal approximation of plane curves", *Computer Graphics and Image Processing*, Vol. 1, pp. 244–256, 1972, doi:10.1016/S0146-664X(72)80017-0

[18] Douglas, D., and Peucker, T., “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", *The Canadian Cartographer*, Vol. 10, pp. 112–122, 1973, doi:10.3138/FM57-6770-U75U-7727

[19] Ester, M., Kriegel, H. P., Sander, J., and Xu, X., “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *2nd International Conference on Knowledge Discovery and Data Mining*, Vol. 96, No. 34, pp. 226–231, 1996

[20] Almeida, J., Barbosa, L., Pais, A., and Formosinho, S., “Improving hierarchical cluster analysis: A new method with outlier detection and automatic clustering", *Chemometrics and Intelligent Laboratory Systems*, Vol. 87, pp. 208–217, 2007, doi:10.1016/j.chemolab.2007.01.005

[21] Rousseeuw, P. J., “Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis", *Computational and Applied Mathematics*, Vol. 20, pp. 53–65, 2009, doi:10.1016/0377-0427(87)90125-7

[22] Davies, D. L., and Donald, W., “A Cluster Separation Measure", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, pp. 224–227, 1979, doi:10.1109/TPAMI.1979.4766909

[23] Dunn, J. C., “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal Of Cybernetics*, Vol. 3, No. 3, pp. 32–57, 1973, doi:10.1080/01969727308546046

[24] Murça, M. C. R., DeLaura, R., Hansman, R. J., Jordan, R., Reynolds T., and Balakrishnan, H., “Trajectory Clustering and Classification for Characterization of Air Traffic Flows", *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016

[25] Murça M. C. R., and Hansman, R. J., “Identification, Characterization, and Prediction of Traffic Flow Patterns in Multi-Airport Systems", *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2018, doi:10.1109/TITS.2018.2833452

[26] Ren, P., and Li, L., “Characterizing air traffic networks via large-scale aircraft tracking data: A comparison between China and the US networks", *Journal of Air Transport Management*, Vol. 67, pp. 181–196, 2018

[27] Enriquez, M., “Identifying temporally persistent flows in the terminal airspace via spectral clustering", *Tenth USA/Europe Air Traffic Management Research and Development Seminar*, 2013

[28] Enriquez, M., and Kurcz, C., “A Simple and Robust Flow Detection Algorithm Based on Spectral Clustering", *5th International Conference on Research and Air Transportation*, 2012

[29] Bombelli, A., Segarra Torné, A., Trumbauer, E., Mease, K. D., “Automated Route Clustering for Air Traffic Modeling", *AIAA Modeling and Simulation Technologies Conference*, 2017, doi:10.2514/6.2017-1318

[30] National PlayBook Website, “`https://www.fly.faa.gov/PLAYBOOK/pbindex.html`", 2018, Accessed: 2018-September-18

[31] Bilimoria, K., Sridhar, B., Chatterij, G., Sheth, K., and Grabbe, S., “FACET: Future ATM Concepts Evaluation Tool", *3rd USA/Europe Air Traffic Management R&D Seminar*, 2000

[32] FAA Website, “`http://aspmhelp.faa.gov/index.php/ASPM_Airports`", 2018, Accessed: 2018-June-20

[33] Ferreira, L., and Hitchcock, D. B., “A Comparison of Hierarchical Methods for Clustering Functional Data", *Communications in Statistics-Simulation and Computation*, Vol. 38, No. 9, pp. 1925–1949, 2009, doi:10.1080/03610910903168603

[34] Matthews, M. P., and DeLaura, R., “Assessment and Interpretation of En Route Weather Avoidance Fields from the Convective Weather Avoidance Model", *10th AIAA Aviation Technology, Integration, and Operations Conference*, 2010

[35] DeLaura, R., and Evans J., “An Exploratory Study of Modeling Enroute Pilot Convective Storm Flight Deviation Behavior", *12th Conference on Aviation Range and Aerospace Meteorology*, 2006

[36] Hartigan, J. A., and Hartigan, P. M., “The dip test of unimodality", *The Annals of Statistics*, pp. 70–84, 1985

[37] Bombelli, A., “Strategic Air Traffic Planning Using Eulerian Route Based Modeling and Optimization", Ph.D. thesis, *University of California, Irvine*, 2017

[38] Cormen, T., Leiserson, C. E., Rivest, R., and Stein, C., “Introduction to Algorithms", MIT press, 2001

[39] Kleinberg, J., and Tardos, E., “Algorithm Design", Pearson Education India, 2006