Journal of Open Aviation Science (2024), Vol.**2**
doi:10.59490/joas.2024.7716

JOAS

ARTICLE

Decentralised Traffic Management for Constrained Urban Airspace: Dynamically Generating and Acting Upon Aggregate Flow Data

Andres Morfin Veytia ,* Joost Ellerbroek , and Jacco Hoekstra

Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology, Delft, Netherlands
*Corresponding author: a.morfinveytia@tudelft.nl

(Received: 1 July 2024; Revised: 28 August and 25 September 2024; Accepted: 25 September 2024; Published: 3 October 2024)
(Editor: Xavier Olive; Reviewers: Patrick Le Blaye, Yash Guleria, Mayara Condé Rocha Murça, Valentin Polishchuk)

Abstract

There are several efforts to explore employing drones to replace ground transportation in cities. However, this would mean that the expected traffic densities would be significantly higher than existing air traffic management. A decentralised system for traffic management may be necessary in this future because (1) not all airspace actors will want to freely share data, (2) the uncertainty of missions due to wind or other factors could make a previous plan inoperable, and (3) the ad hoc nature of urban missions makes them difficult to plan in advance. This work focuses on the challenges of drone operations within constrained urban airspace. We define constrained airspace as a virtual network overlaid on the physical environment, where tall buildings and urban infrastructure dictate the allowed routes. Drones are restricted to flying within this virtual network, either above the existing street network or along other predetermined segments. A dynamic and decentralised traffic management method is presented. The method uses current aggregate flow data to identify and alter the cost of travelling through high-density clusters. The goal is to reduce local traffic density and complexity by encouraging alternate routes. Three different clustering strategies are presented that look at the current position of aircraft and recent safety events. The dynamic traffic management method is first illustrated with two simple example scenarios. Then an experiment is conducted with different traffic demand levels within the city of Rotterdam. It was observed that when using traffic complexity indicators, the method is able to reduce safety events by 30 percent while only increasing the distance travelled by 6 percent.

Keywords: U-space; UTM; conflict resolution; conflict detection; conflict prevention; BlueSky simulator; organic street network; constrained airspace, dynamic traffic management, flow control

Abbreviations: JOAS: Journal of Open Aviation Science

1. Introduction

Ground transportation within urban areas creates congestion, which worsens air quality due to the increased number of vehicles on the road and results in economic losses [1]. A solution may be to transfer a portion of ground transportation to the air, as it has the potential to be more beneficial for the environment [2, 3]. Several government-led research initiatives focusing on drone operations [4, 5, 6, 7] illustrate that there is significant interest in exploring urban drone operations to mitigate issues arising from ground transportation.

An estimate by the European Union predicts 400,000 drones in operation by 2040 [8]. This would re-

© 2024 by the authors. This is an Open Access article, distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) licence (https://creativecommons.org/licenses/by/4.0/)

sult in urban air traffic densities that are significantly higher than traditional air traffic management [9]. Moreover, a significant difference with conventional aviation is that air traffic in cities will need to regularly avoid both dynamic (other aircraft) and static (buildings and geofences) obstacles

In major urban areas, aircraft may need to operate within a constrained airspace, defined here as a virtual network overlaid on the physical environment. This virtual network primarily aligns with and exists above the existing road network, but may also include other predetermined segments. In cities with tall buildings (e.g., New York), flying above the tallest structures could be inefficient. Also, even in areas with shorter buildings, it may still be necessary to fly above the road network, as these paths are typically on public property and may be required by urban regulators. Furthermore, in areas where there are no roads, such as over busy waterways, aircraft may also need to follow the virtual network to ensure predictability. This may help the adoption of drones in cities that do not currently allow flights over busy waterways, such as the Nieuwe Mass River in Rotterdam[10].

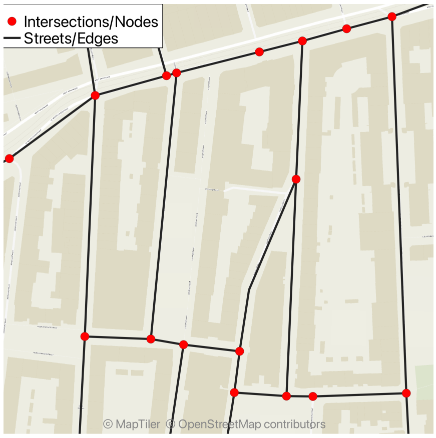


Figure 1. A small virtual network creates a constrained airspace in Rotterdam. The edges of the network are aligned with streets and the nodes are aligned with intersections when the network is above land.

A virtual network creates a constrained airspace that can be described as a graph with nodes and edges. The edges are generally aligned with the streets and the intersections of the edges are nodes (Fig. 1). In this constrained airspace, the manoeuvrability of aircraft is greatly limited. Aircraft are not able to perform heading changes to solve any potential conflicts with other aircraft. Therefore, it is important to have an even spread of traffic over the available airspace to minimize the local traffic complexity and density, as this will lead to a lower conflict probability[11].

The Metropolis II project [12] studied how separation management, flight planning, and airspace structure can be managed in a constrained urban airspace. The project concluded that a hybrid system that combines a central entity, which deconflicts aircraft prior to take-off, and allowing aircraft to perform decentralised conflict resolution was able to combine the benefits of both a centralised and decentralised system.

A question that remains, however, is to what extent central strategic planning is (economically and practically) feasible. It is for instance likely that not all airspace actors will want to freely share operational data, which would be required for central planning. Also, uncertainty of missions due to wind or other factors can make the current plan inoperable. Finally, the ad hoc nature of urban missions makes them difficult to plan in advance in a centralised manner [13]. As such, the current work focuses on a decentralised system in which a set of dynamic rules are incorporated into the current traffic situation.

It has been observed that when following a virtual network [14, 15], drones typically share similar travel legs towards their destination, which creates hot-spots in the airspace and increases the local traffic density and complexity. Traffic complexity attempts to describe the disorder in the airspace based on aircraft interactions [16, 17]. Some measures of traffic complexity try to capture the disorder by observing the proximity and convergence of aircraft [18]. The more convergence present in the airspace, the more unsafe it can become. Additionally, following a virtual network may force aircraft trajectories to converge at the intersections. This makes it a difficult problem to mitigate, especially in a decentralised system.

Previous work in traditional air traffic management [19] created a method for defining dynamic sectors based on the local density. Other works, in urban airspace, used static and historical data to identify zones of high-density traffic [20] and perform capacity management in those zones to reduce local traffic density and complexity. The current work will present a novel dynamic traffic management method that attempts to decentrally reduce local density and complexity by dynamically identifying high-density zones. The method uses real-time aggregate flow information to subdivide the airspace into low and high density zones and applies an additional cost of travel on high density areas.

The dynamic traffic management method can be summarized as follows. (1) Observations of current positions or safety incidents are gathered. The safety incidents are indicators of traffic complexity, and position is an indicator of traffic density. (2) The observations are clustered to create dynamic zones that can receive an additional cost of travel depending on the relative density. (3) Aircraft can then check (decentrally / autonomously) whether their future route intersects these dynamic clusters and update their route taking into account the updated costs. Note that this process happens continuously, the clusters always reflect a recent snapshot of the airspace. This is similar to how highway operators apply speed limits or metering of lanes during rush hour.

The individual agents or drones are responsible for adjusting their own routes that account for the additional cost of travel and find a new optimal path. This will increase the flight distance as aircraft are incentivized to avoid the clustered areas. This creates a trade-off between safety and efficiency when a longer route is chosen.

The dynamic traffic management method will be tested in a simulated urban environment. First, the overall method will be presented along with two example scenarios. The example scenarios are meant to illustrate how the dynamic traffic management method behaves under simplified traffic patterns. Then, the method will be tested in an experiment using a city-wide demand estimation of the city of Rotterdam [21].

2. Method: Using aggregate flow data for dynamic traffic management

This section will outline the method used for performing dynamic traffic management in a constrained urban environment. The method uses aggregate flow statistics to identify and apply additional cost of travel to high-density zones in the airspace. The aggregate flow statistics can be the current position of aircraft or safety events.

Two different but related safety events (conflicts and intrusions) are considered. State-based conflict detection linearly extends the position of aircraft using their current state to check if an intrusion will occur in the near future. A conflict is detected when it is predicted that an aircraft will enter the protected zone of another within a certain look-ahead time. An intrusion occurs when an aircraft actually enters the protected zone of another aircraft.

The difference is illustrated in Figure 2. The protected zone is the dashed circle, in which the radius is the horizontal safe separation distance between aircraft (32 metres, refer to Section 3.2.4 for more information). It is up to the conflict resolution algorithm to ensure that conflicts are solved before they become intrusions. This can be done tactically by performing speed, heading, or altitude changes. In constrained airspace, this is limited to only speed or altitude changes.

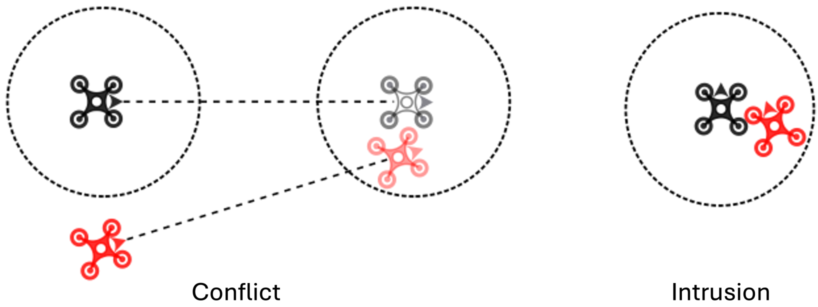


Figure 2. Conflict and intrusion diagram. Note that the protected zone is the dashed circle. The radius of this circle is the safe separation distance between aircraft.

The overall goal of this method is to incentivise aircraft to fly around areas with a high likelihood for conflicts, as this will lead to fewer intrusions and create additional space to tactically solve conflicts. The overall steps of this method are illustrated in Figure 3 and are as follows:

1. Aggregate flow data from urban airspace is gathered into clusters. Figure 3a.
2. The airspace is categorised into a high or low category based on the relative density observed in the clusters. An additional cost of travel is applied to high density zones, Figure 3b.
3. Individual aircraft check if their path will go through areas categorised as high density, Figure 3c.
4. Aircraft find a new optimal plan considering the additional cost of travel of the clusters, Figure 3d.

These steps are continuously repeated over time to provide an updated view of the airspace, occurring every 10 seconds. As a result, different clusters with varying categories are generated every 10 seconds. This 10-second interval comes from the methodology used in [12, 20] to update the densities of high-density zones.

2.1 Gathering aggregate flow data

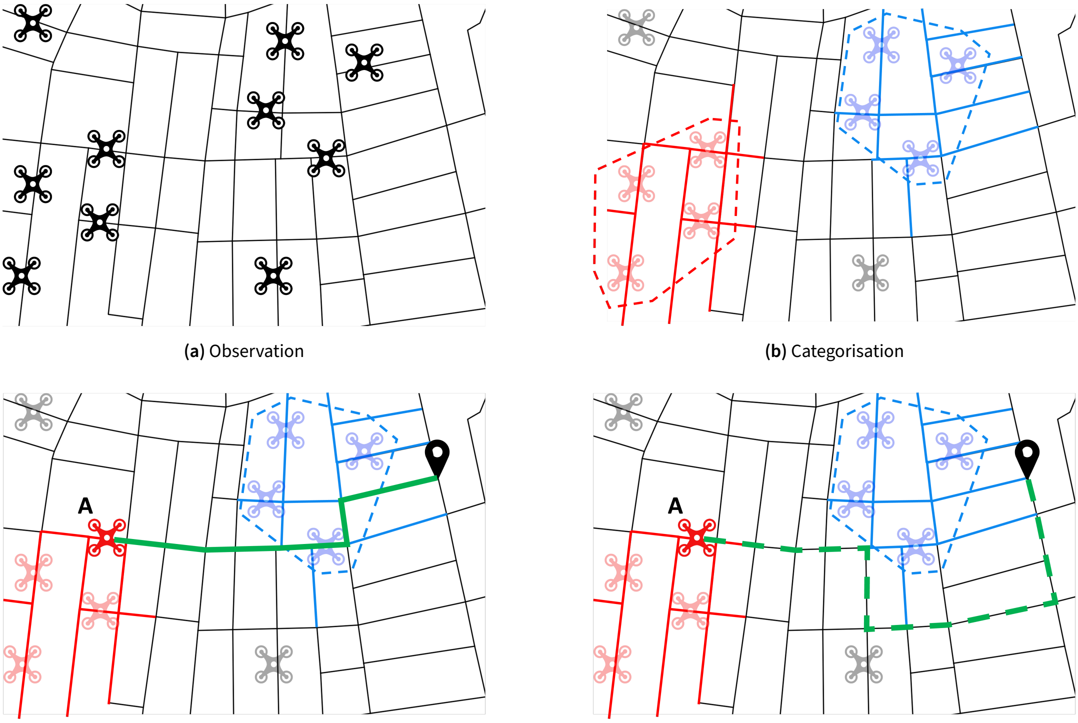
The first step in the method is to make the observation of the aggregate flow data. This work considers three different types of observation strategies to create clusters:

1. Position-based observations: This strategy gathers the current position of all aircraft.
2. Conflict-based observations: This strategy gathers the positions of all conflict events that have occurred in the past ten minutes.
3. Intrusion-based observations: This strategy gathers the position of all intrusion events that have occurred in the past ten minutes.

The observations are gathered into an observation vector, $v_{observation}$:

$$v_{observation} = \begin{bmatrix} (x_1, y_1) \\ (x_2, y_2) \\ .. \\ (x_N, y_N) \end{bmatrix} \tag{1}$$

Each row contains the Cartesian coordinates (x_j, y_j) of each observation and N is the total number of observations. For the position-based strategy N is the number of aircraft in the air. Alternatively,



(a) Observation **(b)** Categorisation **(c)** Original plan **(d)** New plan

Figure 3. The dynamic traffic management method. Note that in this example, the blue cluster is a high-density cluster. Drone A originally has a path that travels through the blue area. However, due to the additional cost of travel applied to the blue area, Drone A creates a new plan that avoids the blue cluster.

for the conflict-based and intrusion-based strategies, the observation vector containts the Cartesian coordinates of aircraft when they entered into a conflict or intrusion, respectively, within the past ten minutes.

The conflict-based and intrusion-based based observations are indicators of traffic complexity. They indicate locations where aircraft may converge (conflicts) or have converged (intrusions). On the other hand, the position-based observations are an indicator of the traffic density and does not include any past information.

2.2 Clustering the observations

In this study, hierarchical clustering, specifically Ward clustering [22], is employed for its distinct advantage over other methods like K-means, which require a predetermined cluster count. Ward clustering involves iteratively merging clusters to minimize the with-in cluster variance during merging. The Ward cluster distance is defined as the Euclidean distance variance when clusters are merged. A Ward distance threshold can be applied to prevent merging of clusters if the variance exceeds this threshold. This approach is similar to the variance-minimizing principle in K-means clustering. Specifically, the Ward cluster distance corresponds to half the squared Euclidean distance between the clusters [23].

$$d_{ward} = \frac{d^2_{euclidean}}{2} \tag{2}$$

This work sets a Ward cluster distance threshold of 4000 m^2, which is about 90 metres, for all three strategies. Note that a sensitivity analysis can be found in the supporting data set [24]. The sensitivity analysis shows how other cluster distances affect the effectiveness of the method. In general, a larger cluster distance corresponds to larger clusters. It was seen in [20] that relatively smaller clusters are preferred in dynamic traffic flow management because it allows for more flexibility to plan around them.

Applying this clustering algorithm to the 2D observation vector will return an array that contains a cluster label for each observation. These clusters are called position-based, conflict-based or intrusion-based clusters depending on the observation vector information.

2.3 Categorising the airspace

After each entry of the observation vector has been placed into a cluster, the airspace can be categorised into areas of relatively lower and higher density airspace. This is illustrated in Figure 3b. Each cluster is represented by a convex hull polygon that contains all observations inside it [25].

2.3.1 Density calculation

The aircraft are operating in a constrained environment in which a virtual network is followed (Fig. 1). Therefore, the density of a cluster is calculated by considering the summed length of the edges that intersect the cluster convex polygon as shown in Equation 3:

$$\text{Linear cluster density} = \frac{\text{Number of observations in cluster}}{\text{Summed length of edges intersecting polygon}} \tag{3}$$

Note that it is possible for clusters to intersect with the same edge or with each other. When this happens, the edge is given the categorisation of the cluster where the edge has its longest segment. Figure 3b illustrates how edges are linked to a convex polygon. The advantage of using 2D space for clustering and 1D space for the density calculation is that the 2D distance matrix is symmetric. This gives a clear indication of the distance between two nodes in the graph. A distance matrix in a one-way fully-connected graph may not be symmetric [26]. For example, the distance from a given node A to node B may not the same as from node B to node A. This makes the graph distance matrix non-trivial for clustering. The reason for choosing a one-way virtual network is that it has been shown to be safer than a two-way virtual network [27].

2.3.2 Apply an additional cost of travel

After the density of each cluster has been calculated, they can be categorised as low- or high-density based on their relative density calculated from Equation 3. For this work, the densities are categorised based on percentiles. The clusters in the lower 25th percentile are labelled as low-density and no cost multiplier is applied. Clusters that are above the 25th percentile are labelled as high-density (refer to the sensitivity analysis for other percentiles).

The unaltered cost of travel is simply equal to the length of the path. At the start of the mission, all aircraft plan their route with the Dijkstra algorithm [28]. The additional cost of travel applies a cost multiplier to the edges that are part of a high-density cluster. The cost multiplier used in the city-wide scenarios is 2 (refer to the sensitivity analysis for effects of the other cost multipliers).

This method ensures that only a current view of the airspace affects the additional cost multipliers, and this is only related to the observation method. An ideal cost multiplier would increase the safety of the airspace while not forcing aircraft to fly needlessly long routes.

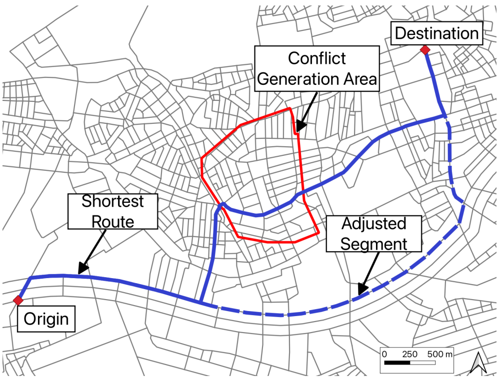
2.4 Checking and modifying the original plan

After the airspace has been categorised, each individual aircraft performs a check to see if their path goes through any high density edges. If the new plan intersects any of the high-density clusters, then a new plan is calculated. The new plan uses the Dijkstra algorithm to calculate an updated path, considering the length and additional cost multipliers.

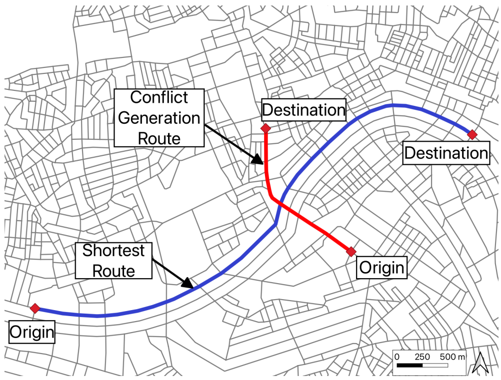
Note that the method acts every ten seconds, therefore, there is a possibility that all aircraft replan at the same time and get into a formation where they all move to the same areas. This oscillation has been observed to happen in road traffic when people use route planning mobile apps like Google Maps [29].

2.5 Example scenarios

This section will illustrate two example scenarios that showcase how the concept behaves in two simple cases. The example scenarios are illustrated in Figure 4. They are simulated in a small section of the Rotterdam virtual network using the BlueSky software (refer to Section 3 for more information about the airspace and software). For a more detailed analysis of the example scenarios, refer to the supporting dataset [24].



(a) Example scenario 1. In this scenario, the aircraft that follow the blue solid line pass through a conflict generation area. The next shortest route that does not pass through the conflict generation area is shown as a dashed line.



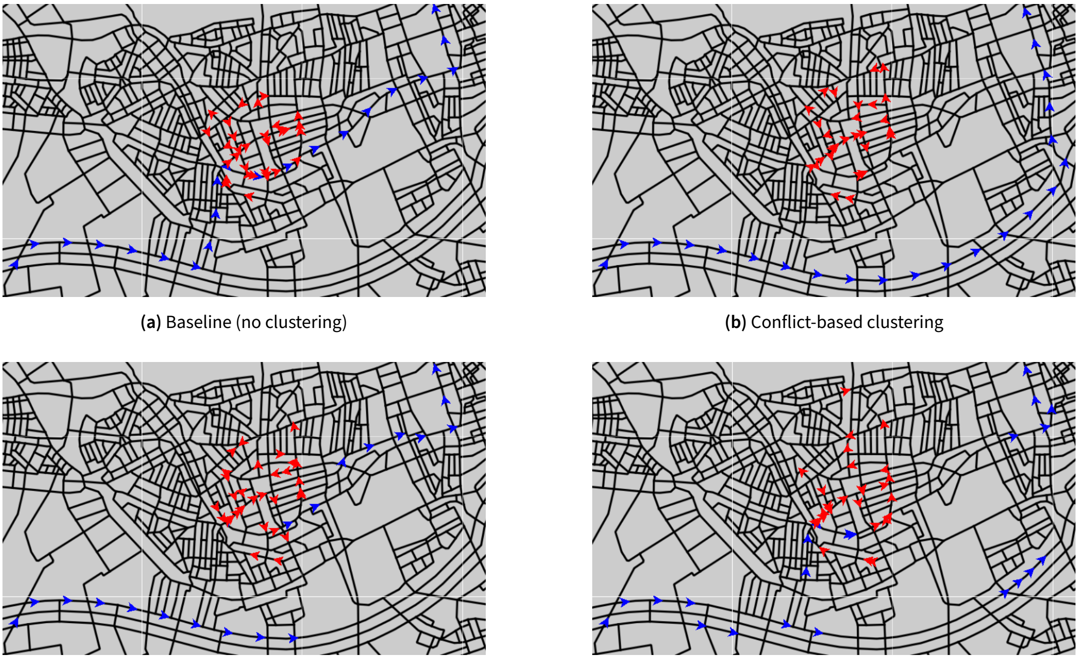
(b) Example scenario 2. In this scenario, the aircraft that follow the blue solid line pass through a conflict generation route. There is no other route that avoids the red line when applying the additional cost multiplier of 1.5.

Figure 4. This figure contains both example scenarios. Example scenario 1 illustrates how the three observations strategies work in the dynamic traffic management method. Example scenario 2 illustrates a scenario in which all additional paths of the blue route go through the red line. This shows that coupling the method with conflict resolution is helpful for cases where the dynamic traffic management method cannot avoid the clustered area.

2.5.1 Example scenario 1

Figure 4a shows the first example scenario. This scenario illustrates how the dynamic traffic management method behaves under the three different clustering strategies. The aircraft following the blue route originally pass through a conflict generation area (red polygon). In the conflict generation area, aircraft spawn and randomly choose a destination within the area. This will create complex traffic in the conflict generation areas as aircraft are converging. The dynamic traffic management method will act and apply an additional cost of travel to observed clusters. The next shortest route that does not pass through the conflict generation area is shown in the blue dashed line.

Figure 5 shows a snapshot of a simulation for the three clustering strategies and a baseline case that does not use the dynamic traffic management method. The red arrows represent individual aircraft flying in the conflict generation area. The blue arrows represent individual aircraft that were



(a) Baseline (no clustering)

(b) Conflict-based clustering

(c) Intrusion-based clustering

(d) Position-based clustering

Figure 5. Simulation snapshot for Example scenario 1. It shows each different observation strategy and a baseline (no clustering) case that does not use the dynamic traffic management method. The red arrows represent individual aircraft flying in the conflict generation area. The blue arrows represent individual aircraft that were initially given the shortest route through the conflict generation area.

initially given the shortest route through the conflict generation area. Figure 5a shows the baseline case. Since the dynamic traffic management method is disabled, the blue aircraft pass through the conflict generation area. Figure 5b shows the conflict-based clustering strategy. Here it is clear that the aircraft in blue are avoiding the high complexity traffic in the conflict generation area. Figure 5c shows the intrusion-based strategy. There are usually fewer intrusions than conflicts, and conflicts between aircraft occur before an intrusion. This is why the blue aircraft in the intrusion-based strategy take both paths to their destination (the shortest route and the adjusted segment). Finally, Figure 5d shows the position-based strategy. The clusters are created using the positions of the aircraft. Therefore, the aircraft in blue are also part of a cluster. This is why some blue aircraft take a parallel route compared to the adjusted segment. These blue aircraft are also avoiding the high complexity area. However, by taking a parallel route, the method is also making the traffic more complex for the blue aircraft.

2.5.2 Example scenario 2

Figure 4b shows the set-up for Example scenario 2. This scenario illustrates how the dynamic traffic management method may not always be able to avoid clusters without taking an excessively long route. In this case, the blue aircraft cross a stream of red aircraft. The scenario is designed so that even with an additional cost multiplier, the blue aircraft cannot avoid the red aircraft. The aim of this example scenario is to only focus on benefits provided to the blue aircraft. The red aircraft in this scenario are not allowed to replan.

Figure 6 shows a snapshot of a simulation for the baseline case and the conflict-based strategy. The red arrows represent individual aircraft flying the conflict generation route. The blue arrows repre-

(a) Baseline (no clustering)

(b) Conflict-based clustering

Figure 6. Simulation snapshot for Example scenario 2. This shows the baseline (no clustering) case and the conflict-based clustering strategy. The red arrows represent individual aircraft flying the conflict generation route. The blue arrows represent individual aircraft that were initially given the shortest route through the conflict generation route.

sent individual aircraft that were initially given the shortest route through the conflict generation route. Figure 6a shows that the blue aircraft in the baseline case do not change their route in response to the red aircraft. Figure 6b shows the conflict-based clustering strategy. It is clear that some aircraft in blue are taking a different route compared to the shortest routes. These new routes are in response to the clusters created by the observed conflicts. However, this route does not avoid crossing the red aircraft. The traffic complexity of the scenario is not simplified by the dynamic management method. Nevertheless, the dynamic traffic management method is spreading out the blue traffic to parallel edges, thereby reducing the local density at the intersections. If there is a robust conflict resolution strategy, the additional space created by the dynamic traffic management method may improve the effect of the conflict resolution algorithm.

3. Experimental design

The previously presented method aimed to identify and mitigate traffic hotspots to reduce the number of safety incidents (conflicts and intrusions). By dynamically redistributing aircraft, the local traffic density and complexity may be reduced, and more space can be provided for conflict resolution.

An experiment was designed to test the dynamic traffic management method under several city-wide scenarios with varying traffic demand level. The three clustering strategies will be compared against each other and to the baseline case that does not use the method.

3.1 Hypotheses

H1: It is hypothesized that all three clustering strategies will have higher safety levels than the baseline (no clustering). The method is expected to be able to identify locations with conflict, intrusion, and positional hot-spots.

H2: It is hypothesized that traffic complexity clusters (conflicts and intrusions) will have higher safety levels than density clusters (position). This is because complexity clusters will identify sectors where aircraft are converging. On the other hand, density clusters may increase the cost of travel to areas with a relative high traffic density but low complexity.

H3: It is hypothesized that the intrusion-based clustering will have higher safety levels than the conflict-based clustering strategy. The usage of state-based conflict detection in constrained airspace has been known to create false conflicts. Therefore, using conflicts may increase the cost of travel in more low traffic complexity areas as compared to clustering with intrusions.

These three hypotheses will be evaluated with a Wilcoxon signed-rank test [30]. The null hypothesis is that the differences between strategies is not statistically significant, therefore the listed hypotheses will be accepted when $p_{value} < 0.01$. Refer to Appendix 1 for the detailed results of the hypothesis testing.

3.2 Common Elements

The common elements of the experiment will be shown in the following section. They include the urban airspace, the traffic demand patterns, origins and destinations, the conflict resolution algorithm, aircraft models, and the simulation software.

3.2.1 Rotterdam Airspace

This work simulates air traffic in the centre of Rotterdam. The Rotterdam airspace is particularly interesting because the city is traversed by the Nieuwe Maas river, see Fig. 7. The airspace illustrates that when the virtual network is over land, it is aligned with the street network. Additionally, in Rotterdam there are a limited number of land bridges connecting north and south. Therefore, several new drone "bridges" have been proposed to ease congestion and ensure that the graph is well-connected (all nodes can reach all other nodes). Currently, the government of the Netherlands does not allow any drone flights over the river due to maritime operations [10]. Therefore, enhancing the predictability of drones over waterways could lead to increased acceptance over these areas.

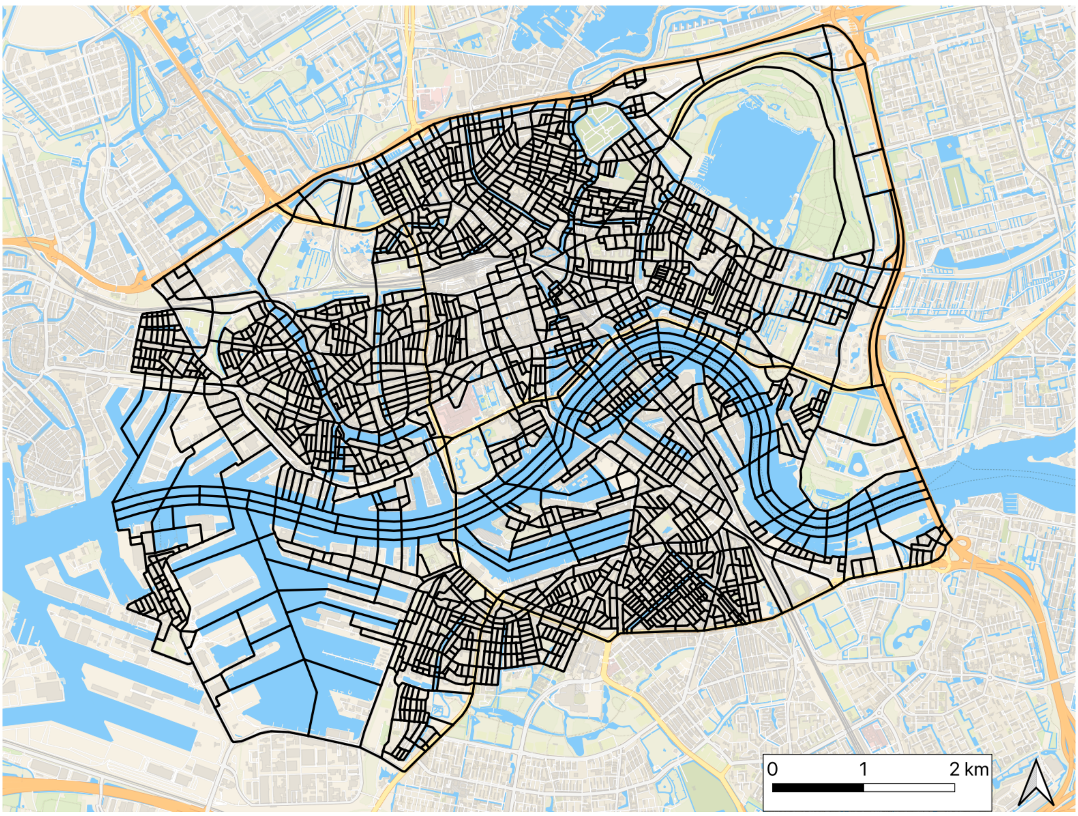


Figure 7. Rotterdam airspace

The process of adapting a street network to a virtual network is derived from [14]. The street network is download from OpenStreetMap [31] with osmnx [32]. After downloading, the street network is simplified by removing parallel streets, roundabouts, and dead-ends. Following the simplification, several edges are added to the virtual network that are in parallel to waterways. The distance between these parallel edges is three times the safe separation distance of aircraft. This is about the distance that an aircraft would travel with their average speed for the conflict look-ahead time of 10 seconds. Finally, the edges above land that are dead-ends are extended to the ones parallel to the waterways.

This work assumes that all edges are one-way, since it is generally safer than two-way edges [27]. However, in a one-way non-orthogonal network, it is not apparent what the direction of each edge should be to ensure that all intersections are reachable from each other. The process of choosing the direction of the edges is described in detail in [14] and [12] and involves two important steps.

The first step is to group the edges so that edges in the same group are continuous at intersections. This is done with the Continuity in Street Networks algorithm [33] which uses the edge geometry to assign the groups based on continuity. The second step is to use a genetic algorithm to decide the direction of each group. The goal of the genetic algorithm is to (1) create a network that is strongly connected, and (2) find a suboptimal solution where the overall cost of travel between all intersections is minimized, and (3) ensure that head on situations are not created as those create high relative velocities which lead to decreased safety [34]. The result is a one-way and fully-connected virtual network with intersections that do not create head-on collisions. The final network used in this work is seen in Fig. 7.

Note that, the traffic management system proposed in Sec. 2 should be independent of the virtual network that it operates in. Therefore, the sensitivity analysis [24] shows how the method performs in the virtual network of Vienna, from [12], in which the virtual network is always above the existing street network.

3.2.2 Traffic demand pattern

This section will present the results of testing the method on city-wide traffic scenarios. The MASS-GT [21, 35] project made an estimate of the daily parcel demand for South Holland. With this data, each neighbourhood of Rotterdam can be given a demand percentage relative to other neighbourhoods. The relative demand of each neighbourhood can be seen in Fig. 8.

3.2.3 Origins and destinations

About two-hundred random intersections are selected as possible take-off locations (Fig. 8), they are at least 300 metres away from each other. Then for each possible origin, a path is created to all other intersections in the network with the following constraints. For each mission, the origin and destination are at least 1000 metres apart, and it cannot start or end in a node that is inside water. This creates around 500.000 possible routes. Each route is created with the Dijkstra algorithm [28] using the length as a cost to select the shortest route.

To create the scenarios, random origins are selected from the possible take-off locations, then a destination is selected considering the weighted probabilities of the destination nodes based on the parcel demand. All missions take place at 30 feet above the ground.

3.2.4 Conflict detection and resolution

This work uses state-based conflict detection. This method linearly extrapolates the current position of all aircraft with a given look-ahead time and checks if they will violate the protected zone (Fig. 2). A known effect of using state-based conflict detection in non-orthogonal constrained airspace is that it may detect false conflicts [14]. These are detected conflicts that do not actually become intrusions.

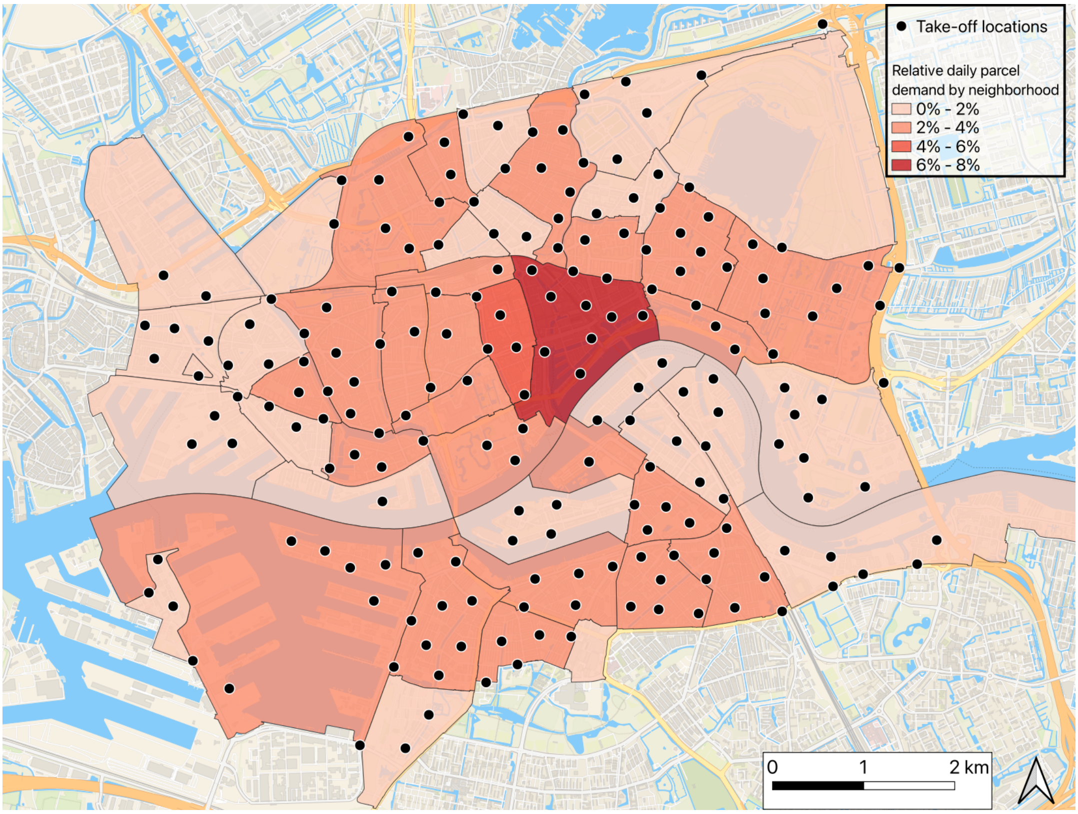


Figure 8. Rotterdam parcel demand with the possible take-off locations.

The drones in this work use a tactical speed-based conflict resolution algorithm from [36] to solve conflicts in constrained airspace. This method has a horizontal protected radius of 32 metres based on the horizontal accuracy signal-in-space requirements (Table 3.7.2.4-1) [37]. The work uses a 10 second lookahead time that was used in previous urban airspace research works [14, 12, 20, 36]. Note that the tactical conflict resolution is independent of the dynamic traffic management method, as it only forces aircraft to slow down or speed up.

3.2.5 Aircraft models

This work used a performance model based on the DJI Matrice 600 Pro hexacopter [38]. The model parameters are shown in Table 1.

Table 1. Performance model parameters.

Parameter	Value
Max. horizontal speed	12.9 m/s
Avg. horizontal speed	10.3 m/s
Min. horizontal speed	0 m/s
Max. take-off mass	15 kg
Acceleration/Deceleration	3.5 m/s^2

3.2.6 Simulation software

This work used the BlueSky air traffic simulator for the experiments [39]. BlueSky is a fast-time simulator that can be extended via plugins so that a cluster observation strategy and flight planning module can be added. This work also made use of the Metropolis II developments for BlueSky to improve the capabilities of simulating traffic in urban environments [12].

3.3 Independent variables

- Dynamic traffic management method clustering strategy: baseline (no clustering), position-based, conflict-based, intrusion-based.
- Imposed traffic demand level: 100, 200, 300, 400, 500 instantaneous aircraft.

Moreover, each case is repeated 20 times with a different random seed that generate different origin-destination pairs. This creates 4 (clustering strategies + baseline) x 5 (traffic demand levels) x 20 (randomly selected random seeds) = 400 different scenarios. Each scenario has a simulation time of 2 hours. The work was simulated on an AMD Ryzen 9 5950X 16-Core Processor with 96 GB of RAM. It took about 25 hours of real time to simulate 800 hours (400 scenarios * 2 hours).

3.4 Dependent variables

This section will present the dependent variables used in the city-wide scenario experiment. There are three different overall categories of dependent variables (safety, efficiency and clustering). Note that the results of the first 10 minutes are ignored because that is the observation time required for the conflict-based and intrusion-based clustering strategies.

3.4.1 Safety: conflicts and intrusions

The safety metrics used to evaluate the results are conflicts and intrusions. However, these are not presented as absolute values. Note that because the number of intrusions and conflicts scales with the separation distance and look-ahead time, it is more useful to consider the relative difference between concepts when comparing them.

The metrics are presented in terms of conflicts and intrusions per flight. These are calculated by dividing the total amount of conflicts and intrusions by the amount of flights. Additionally, the safety metrics are also presented per distance travelled as a percentage of the baseline (no clustering). For example, assume that the conflict-based strategy shows 105 conflicts per distance percentage. This indicates that aircraft in this clustering strategy have 5 percent more conflicts per distance travelled than aircraft travelling in the baseline.

Finally, a heat-map of the intrusions is also presented to show how the clustering strategy visually change the traffic patterns. It uses the kernel density estimation [40] with a radius of 300 metres.

3.4.2 Efficiency: distance travelled and number of replans

The distance travelled percentage is presented as a percentage of the total distance travelled by aircraft in the baseline.

The number of replans per flight is also presented. A replan generally indicates that the aircraft took a longer route. Note that by definition, aircraft in the baseline case do not make any replans. Note that an aircraft may receive a flight plan that it does not execute because a newer plan is issued before the previous update is implemented. Consequently, the number of replans per flight reflects only the executed updates, not the attempts.

3.4.3 Clustering: cluster percentage and stability

Since there is a new observation vector made every 10 seconds (Section 2), the clusters at one time may differ from the clusters at the next time step. If the clusters change significantly, this may affect the stability of the airspace, because there will be less coherence in replanning.

Therefore, two metrics are presented. The first is the percent of clustered airspace. This shows on average how much of the airspace is part of a cluster during each observation. The second is the cluster temporal stability. This represents the average percentage overlap of the clusters from one observation to the next. A value of 100 percent would indicate that the cluster polygons are completely identical from one observation time step to the other.

Finally, an example map of the clusters with a high density categorisation at 400 aircraft for each of the three strategies is presented to help visualise the method.

4. Results

This section presents the results of the city-wide scenarios. Unless noted otherwise, the box plots show the metric on the vertical axis and the imposed traffic demand level on the horizontal axis.

4.1 Safety: Conflicts and intrusions

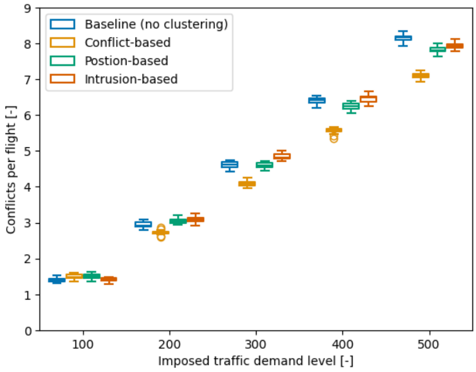
Fig. 9a shows the conflict occurrence per flight. It shows that the number of conflicts per flight increase with traffic demand level for all cases. Also, the conflict-based strategy resulted in the lowest number of conflicts per flight, starting from 200 aircraft. The position-based and intrusion-based strategies both have a number of conflicts per flight that is similar to the baseline (no clustering) at most demand levels, except at 500 aircraft, where they are slightly lower.

Fig. 9b shows the number of intrusions per flight. This plot shows that the number of intrusions per flight increase with traffic demand level. Moreover, the conflict-based strategy resulted in the lowest number of intrusions per flight at most traffic demands. The intrusion-based and position-based clustering strategies are only better than the baseline, starting at 400 aircraft. At these higher demands, the intrusion-based clustering strategy is slightly better than the position-based strategy.

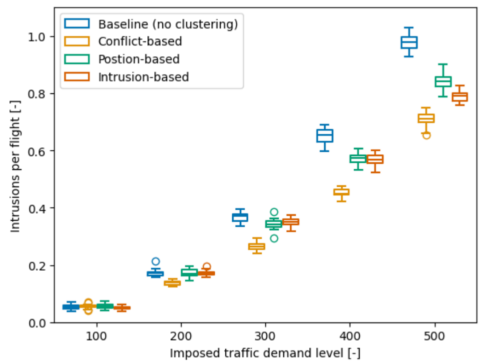
Fig. 9c shows the occurrence of conflicts per distance as a percentage of the baseline. The baseline is illustrated by a dashed line at 100 percent. Similar to the previous plots, the conflict-based strategy has the lowest conflicts per distance percentage. Reaching the maximum improvement of 85 percent at 300 aircraft. The position-based and intrusion-based strategies are slightly below 100 percent at 300 aircraft, but are not able to reach the level of the conflict-based strategy.

Fig. 9d shows the occurrence of intrusions per distance as a percentage of the baseline. This plot shows that the conflict-based strategy has a 70 percent improvement from the baseline at most traffic demand levels. The position-based and intrusion-based strategies have similar values and show improvement over the baseline case at 300 aircraft. Again, they are not able to reduce the number of intrusions per distance travelled as much as the conflict-based strategy.

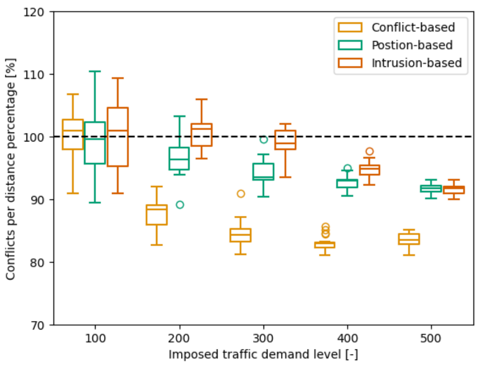
Fig. 10 shows a cumulative intrusion map for one scenario of 400 aircraft in the air. Figs. 10a, 10b, show the baseline and conflict-based, strategies, respectively. It can be seen that the baseline case has noticeably brighter hot-spots relating to a higher number of intrusions. Note that the position-based and intrusion-based clustering strategies had a similar map as the conflict-based, so they are not shown here. (refer to the supporting dataset [24]).



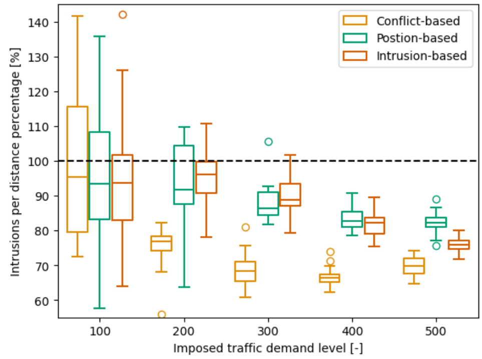
(a) Conflicts per flight.



(b) Intrusions per flight.



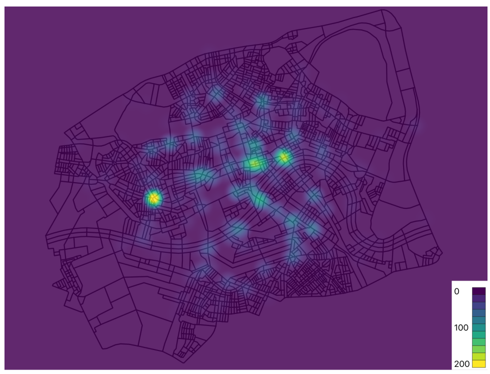
(c) Conflicts per distance percentage.



(d) Intrusions per distance percentage.

Figure 9. Safety: These plot show the conflicts and intrusions per flight in Fig. 9a and Fig. 9b, respectively. It also shows the conflicts and intrusions per distance percentage in Fig. 9c and Fig. 9d, respectively.

(a) Baseline (no clustering)



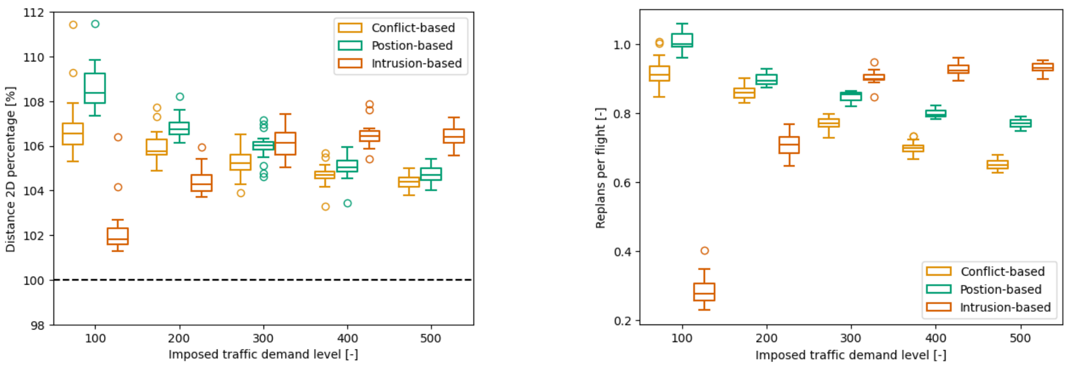
(b) Conflict-based clustering

Figure 10. This figure shows the cumulative intrusion maps for one of the 400 aircraft scenarios for the baseline (no clustering) case and the conflict-based clustering strategy. The colours of the heat map represent the number of intrusions at that location.

4.2 Efficiency: Distance travelled and replans

Fig. 11a shows the distance travelled as a percentage of the baseline case. The plot shows that all clustering strategies force aircraft to travel longer than the baseline. However, this increase never exceeds 10 percent of the baseline value. The intrusion-based clustering strategy forces aircraft to travel the least amount of distance at 100 and 200 aircraft as compared to the other strategies. Interestingly, it has the highest distance travelled at 400 and 500 aircraft. The conflict-based strategy has slightly lower distances than the position-based at most traffic demand levels. The conflict-based strategy is also the lowest at 400 and 500 aircraft.

Fig. 11b shows the number of replans per flight. The position-based clustering strategy has the highest number of replans per flight at low demand levels. Conversely, at the low demand levels, the intrusion-based strategy has the lowest number of replans per flight. However, starting at 300 aircraft in the air, the conflict-based strategy has the lowest number of replans per flight, while the intrusion-based has the highest. It is also interesting to note that the conflict-based and position-based strategies show a similar decreasing trend of replans per flight. On the other hand, the replans per flight in the intrusion-based strategy increases from 100 to 300 aircraft and then tapers at 400 and 500 aircraft.

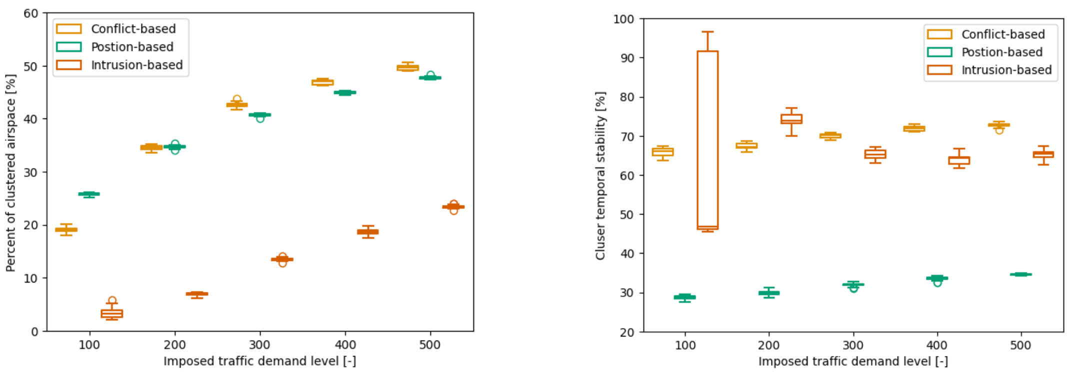


(a) Distance as percentage of the Baseline case.

(b) Replans per flight.

Figure 11. Efficiency: These plot show the distance percentage and amount of replans per flight in Fig. 11a and Fig. 11b, respectively.

4.3 Clustering: Cluster percentage and stability

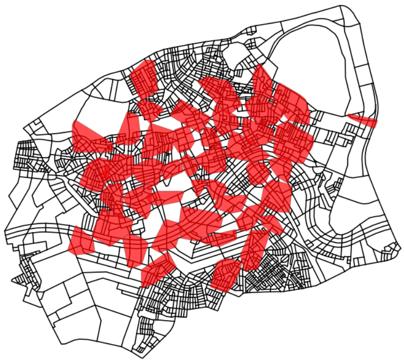


(a) Percent of clustered airspace.

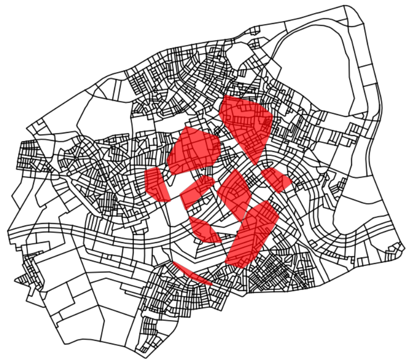
(b) Temporal cluster stability.

Figure 12. Clustering: These plots show the percent of clustered airspace and temporal cluster stability in Fig. 12a and Fig. 12b, respectively.

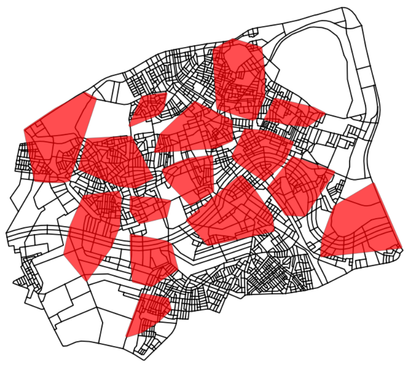
Fig. 12a shows the percent of the airspace that is part of a cluster. Note that at all traffic demand



(a) Conflict-based clusters with a high density categorisation at 50 minutes of a simulation with 400 aircraft.



(b) Intrusion-based clusters with a high density categorisation at 50 minutes of a simulation with 400 aircraft.



(c) Position-based clusters with a high density categorisation at 50 minutes of a simulation with 400 aircraft.

Figure 13. Clustering: These maps show the cluster with a high density at 50 minutes of a simulation with 400 aircraft for the three strategies.

levels, the intrusion-based strategy has the least amount of airspace in a cluster. Varying from 5 percent to a maximum value of around 20 percent. At 100 aircraft, the position-based strategy has the highest amount of clustered airspace. However, at higher demand levels the conflict-based strategy tends to have more clustered airspace, up until almost 50 percent at 500 aircraft.

Fig. 12b shows the cluster temporal stability. The results show that the intrusion-based strategy has a large amount of variation at 100 aircraft, this is because there are few intrusions at this density. It then stabilizes at around 65 percent for higher demand levels. The temporal stability of the conflict-based strategy slightly increases, with traffic demand reaching around 70 percent. The temporal cluster stability of the position-based strategy is the lowest for all traffic demand levels, with a maximum of 35 percent.

Fig. 13 shows clusters categorised as high density, at 50 minutes of a simulation with 400 aircraft in the air. Note that these plots only show a single moment in time so they should be analysed in conjuction with Figs. 12a and 12b. Fig. 13a shows that the conflict-based clusters are concentrated in the middle of the airspace. Fig. 13b shows that there significantly less clusters identified with the intrusion-based strategy. Finally, Fig. 13c shows that the position-based clusters tend to spread over more of the airspace.

5. Discussion

The dynamic traffic management method aimed to improve the overall safety of the airspace by applying an additional cost of travel to high-density clustered zones. These clustered zones are created and continually updated by using current observations of positions, conflicts, and intrusions. Each individual aircraft checks if it will intersect clustered areas and makes a new plan to avoid them. The goal is for aircraft to avoid areas with high local traffic density and complexity.

Hypothesis **H1** can only be partially accepted because the clustering strategies did not show improvements over the baseline at all traffic demand levels. When comparing the baseline to the conflict-based clustering strategy, the safety improvements over the baseline case are only significant starting at 200 aircraft in the air ($p_{value} < 0.01$). The intrusion-based and position-based strategies are only better than the baseline, starting at 300 aircraft in the air. These results indicate that the method's effectiveness depends on the presence of sufficient safety event data to form meaningful clusters.

Interestingly, the method is able to limit the extra distance travelled to less than 110 percent of the baseline case. At higher demand levels, when the method is effective for all cluster strategies, the extra distance travelled is around 106 percent of the baseline, while the intrusions per distance percentage is on average lower than 90 percent of the baseline. This shows the trade-off made in the dynamic management method between efficiency and safety.

The intrusion maps show that the clustering strategies are able to dissipate the intrusion hot-spots seen by the baseline case. At higher demand levels, all the strategies are better than the baseline case. The method has achieved the goal of dissipating the intrusion hot-spots without creating new ones due to the replanning at high demand levels. The supplemental documents [24] show the other heat-maps and some animations of the traffic.

Additionally, the results highlight the reactive nature of the dynamic traffic management method. The airspace must have clusters of positions or safety events to be able to solve them. Clusters mean that aircraft are close to each other, which then leads to conflicts and intrusions. A strategic approach might be able to solve these issues before they happen. However, for a strategic planner to work, the flight plans will need to be provided to a central entity to deconflict them prior to take-off. This could make it more difficult to deal with uncertainties such as wind. Moreover, if urban air traffic is meant to replace a portion of ground operations, the dynamic and ad-hoc nature of ground operations will also add some complexity to the strategic planner.

Hypothesis **H2** can be partially accepted. The conflict-based clustering strategy has higher safety levels than the position-based strategy, starting at 200 aircraft in the air, ($p_{value} < 0.01$). However, the intrusion-based strategy only performs better than the position-based at 500 aircraft in the air. The clusters that take into account traffic complexity tend to have higher safety levels than those that consider traffic density for clusters. This can be seen in Example scenario 1 which shows that the position-based clustering forces aircraft to replan even when they are in a relatively low traffic complexity situation. However, it can also be seen that as the traffic demand level increases, the position-based clustering moves closer to the safety level of the conflict-based clustering.

It was observed that the aircraft replans in the position-based clusters has a similar trend than that of the conflict-based concepts. However, these replans are not as effective as reducing the number of intrusions as in the conflict-based strategy. This can be explained by two reasons: (1) the percent of clustered airspace is relatively high (maximum at 50 percent) and the cluster temporal stability is relatively low (maximum 30 percent). Therefore, the location of the clusters is not generally overlapping, causing some oscillatory effects that change the locations of the high-cost airspace. Also, because the traffic demand pattern is not uniform, the location of the high traffic density areas will vary, making traffic density a less effective clustering strategy than traffic complexity. Note that for the intrusion-based strategy, there is only a statistically significant difference with the position-based strategy at 500 aircraft in the air. Therefore, complexity metrics that indicate locations where aircraft may converge (conflicts) are generally more effective than traffic density clustering strategies. The conflict-based complexity indicator is less likely to cluster areas where there is a high traffic density but aligned aircraft (Example scenario 1).

Hypothesis **H3** is rejected. The results show that the conflict-based clustering strategy achieved a higher level of safety than the intrusion-based strategy, starting at 200 aircraft in the air. While the null hypothesis is rejected, meaning there is statistically significance when comparing these two strategies ($p_{value} < 0.01$), the trend is the opposite to what was hypothesized. It has been shown that using state-based conflict detection in constrained urban airspace tends to create false conflicts (conflicts that never become intrusions) [14]. However, the conflict-based strategy is more effective at avoiding areas with higher traffic complexity and density than the intrusion-based strategy at most traffic demands.

At lower demand levels, the number of intrusions per flight is an order of magnitude less than the conflicts per flight. Therefore, for the intrusion-based strategy, the method does not have enough information to avoid complex traffic areas. This is also why the distance travelled and replans per flight is initially low for the intrusion-based strategy, there are not many intrusion events that create clusters and force aircraft to replan at the low demand levels. At higher demand levels, there is enough information about complex traffic areas, since there are many more intrusions. This is why the intrusion-based strategy is reaching the safety levels of the conflict-based strategy.

The conflict-based strategy has both a high percent of clustered airspace and a high temporal stability. This means that the conflict-based strategy limits the oscillatory effects seen in the position-based strategy and has enough clustered airspace to redistribute aircraft through lower cost routes. Note that the highest percentage of clustered airspace is never more than 50 percent of the total airspace. In contrast, the intrusion-based strategy has a very low amount of clustered airspace at low demand levels. This is also why aircraft travel the least amount of distance and replan the least amount of times at these low densities. There is simply not enough data to identify the high traffic complexity areas. However, as the traffic demand level increases, and there are more intrusions events, the replans per flight are the highest for the intrusion-based strategy. At these high demand levels, the intrusion-based strategy has higher safety levels than the position-based strategy and is approaching the level of the conflict-based strategy.

It is interesting to note that the intrusion-based strategy has a downward trend for the conflicts and intrusions per distance percentage. In contrast, the conflict-based strategy achieves its lowest relative gain of intrusions per distance at 200-300 aircraft. With increasing demand levels, the conflict-based strategy is not able to decrease more than 70 percent. While it shows that the conflict-based strategy is robust with the increasing demand level, it is possible that the false conflicts detected are making it difficult to keep improving.

6. Conclusion

It was observed that the dynamic traffic management method improves the safety of the airspace. The maximum improvement reduced the intrusions per distance to 70 percent of the baseline (no clustering). It accomplishes this without suggesting excessively long alternate routes. The distance travelled was always less than 10 percent higher than the baseline, and around 6 percent for the best performing strategy (conflict-based).

The work shows that using different aggregate flow data strategies has varying success in avoiding intrusions between aircraft. The clustering strategies that take into account traffic complexity (conflict-based and intrusion-based) were more successful than considering only traffic density (position-based). Amongst the complexity cluster strategies, the conflict-based strategy was the most successful at reducing the number of observed intrusions while limiting extra distance travelled. It is better performing than the intrusion-based strategy because it provides more information about complex traffic areas. This also means that even false conflicts, which are generally seen as unfavourable, can have some beneficial effect. However, they may also limit the performance of the method when the traffic demand levels are high. It is also possible to create a traffic complexity metric by smoothing the traffic density over a certain time limit. This adjusted position-based strategy may mitigate the issues caused by false conflicts.

The proposed method is able to show that a decentralised traffic management system is able to improve upon a baseline (no clustering) case, especially when using traffic complexity metrics. However, this should be compared against other decentralised capacity management system such as those in [20] that use static zones. Additionally, the reactive nature of the method means that there must be a certain number of safety events observed to work effectively. This is illustrated by the improved safety performance in the intrusion-based strategy with traffic demand level. Therefore, research that combines this strategy with a more strategic planner could further improve the safety.

Further research should focus on trying to improve the understanding of this method. Mainly, it can focus on studying how the parameters, such as additional cost and cluster distance threshold, affect the effectiveness of the method on different virtual networks. It is also possible to further refine how the zones are categorised. Currently, a convex polygon is used to categorise groups of edges. This means that some edges without any traffic may be categorised as high density. However, it might be better to only consider edges where aircraft are actually flying and creating conflicts or intrusions. Moreover, rather than having two airspace categories, making the additional cost multiplier proportional to the measured density of the edges may improve the method.

Moreover, data transmission latency was not considered in this work. It should be studied in concurrence with the cluster update rate to learn the feasibility of the level of communication required. As it was observed that not all aircraft actually make new plans, further research can also focus on limiting the ratio of aircraft that are allowed to replan. It might be that only a certain number of aircraft need to alter their plans in order to see a beneficial effect on safety.

Appendix 1. Hypothesis testing results

This appendix will illustrate the formal hypothesis testing results for the hypotheses presented in Section 3.1. The safety levels of the clustering strategies and baseline case will be compared based on the observed intrusions per flight with a Wilcoxon signed-rank test [30]. The intrusions per flight metric is chosen as the relevant metric because it represents aircraft converging at distances less than the safe separation distance.

The null hypothesis is that the differences between strategies is not statistically significant, therefore the null hypotheses will be rejected when $p_{value} < 0.01$

Appendix 1.1 H1 hypothesis test

Hypothesis **H1** states that all three clustering strategies will be safer than the baseline case. The null hypothesis is that the baseline case is not statistically different from any of the three clustering strategies. The p_{value} comparing the baseline case to the three clustering strategies is seen in Table 2. Each column represents how that particular strategy compares to the baseline. Hypothesis **H1** was partially accepted because the clustering strategies only have higher safety levels at the higher traffic demand levels.

Table 2. p_{values} comparing the baseline (no clustering) case to the three clustering strategies. Highlighted cells refer to the cases where the null hypothesis can be rejected.

	Clustering strategies compared to the baseline (no clustering) case		
Traffic demand level	**Conflict-based p_{value}**	**Intrusion-based p_{value}**	**Position-based p_{value}**
100	0.5	0.23	0.41
200	<0.01	0.89	0.72
300	<0.01	<0.01	<0.01
400	<0.01	<0.01	<0.01
500	<0.01	<0.01	<0.01

Appendix 1.2 H2 hypothesis test

Hypothesis **H2** states that two traffic complexity clustering strategies (conflict-based and intrusion-based) will have higher safety levels than the traffic density strategy (position-based). The null hypothesis is that the position-based strategy is not statistically different from the conflict-based and intrusion-based strategy. The p_{value} comparing the position-based case to the two traffic complexity clustering strategies is seen in Table 3. Each column represents how that particular strategy compares to the position-based strategy. Hypothesis **H2** was partially accepted because the conflict-based strategy has higher safety levels than the position-based, starting at 200 aircraft in the air.

Table 3. p_{values} comparing the position-based strategy to the two traffic complexity strategies (conflict-based and intrusion-based). Highlighted cells refer to the cases where the null hypothesis can be rejected.

	Clustering strategies compared to the position-based case	
Traffic demand level	**Conflict-based p_{value}**	**Intrusion-based p_{value}**
100	0.65	0.01
200	<0.01	0.84
300	<0.01	0.06
400	<0.01	0.67
500	<0.01	<0.01

Appendix 1.3 H3 hypothesis test

Hypothesis **H3** states that the intrusion-based strategy will have higher safety level than the conflict-based strategy. The null hypothesis is that the strategies are not statistically different from each other. The p_{value} comparing both complexity strategies is seen in Table 4. Hypothesis **H3** was rejected because the conflict-based strategy had a higher level of safety at most traffic demand levels.

Table 4. p_{values} comparing the conflict-based strategy and intrusion-based strategies. Highlighted cells refer to the cases where the null hypothesis can be rejected.

Traffic demand level	p_{value}
100	0.03
200	<0.01
300	<0.01
400	<0.01
500	<0.01

Acknowledgement

The authors would like to thank their reviewers for all their helpful and insightful comments that led to the improvement of this work. Moreover, we would like to thank Calin Andrei Badea for being a helpful outside eye when finding programming bugs.

Author contributions

- Andres Morfin Veytia: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Project Administration, Software, Visualization, Writing (Original Draft), Writing (Review and Editing).
- Joost Ellerbroek: Supervision, Software, Writing (Review and Editing), Project Administration
- Jacco Hoekstra: Supervision, Software, Writing (Review and Editing), Project Administration.

Open data statement

The accompanying sensitivity analysis, software, scenario logs, and additional figures set are provided in [24][1]. We have also created a small video demonstration of the method here.

Reproducibility statement

The method for reproducing the result is explained in the supplementary repository [24]. In summary, the process is as follows:

1. Install the python packages required by BlueSky [41] and those required by this paper. There is a list of packages provided in the repository.
2. Use the BlueSky source code provided in the repository.
3. Run the scenarios provided in the repository. The HOWTOSCENARIOS.md document explain how to run scenarios in detail.
4. Create the plots with the logs created by the scenarios. The HOWTOCREATEPLOTS.md document explains how to use the post-processing python code to generate the plots in this paper and some additional ones.

References

[1] Felix Creutzig, Aneeque Javaid, Nicolas Koch, Brigitte Knopf, Giulio Mattioli, and Ottmar Edenhofer. "Adjust urban and rural road pricing for fair mobility". In: *Nature Climate Change* 10.7 (July 2020), pp. 591–594. ISSN: 1758-6798. DOI: 10.1038/s41558-020-0793-1. URL: https://doi.org/10.1038/s41558-020-0793-1.

[2] Joshuah K. Stolaroff, Constantine Samaras, Emma R. O'Neill, Alia Lubers, Alexandra S. Mitchell, and Daniel Ceperley. "Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery". In: *Nature Communications* 9.1 (Feb. 2018), p. 409. ISSN: 2041-1723. DOI: 10.1038/s41467-017-02411-5. URL: https://doi.org/10.1038/s41467-017-02411-5.

[3] Aishwarya Raghunatha, Emma Lindkvist, Patrik Thollander, Erika Hansson, and Greta Jonsson. "Critical assessment of emissions, costs, and time for last-mile goods delivery by drones versus trucks". In: *Scientific Reports* 13.1 (July 2023), p. 11814. ISSN: 2045-2322. DOI: 10.1038/s41598-023-38922-z. URL: https://doi.org/10.1038/s41598-023-38922-z.

[1] https://doi.org/10.4121/54825f14-8743-447d-8346-3afa46d319d6

[4] DLR. *DLR Blueprint Concept for Urban Airspace Integration.* DLR, 2017. URL: https://www.dlr.de/de/medien/publikationen/sonstige-publikationen/2017/blueprint-concept-for-urban-airspace-integration_2933.

[5] Single European Sky ATM Research 3 Joint Undertaking. *U-space : blueprint.* Publications Office, 2017. DOI: 10.2829/335092.

[6] Parimal Kopardekar, Joseph Rios, Thomas Prevot, Marcus Johnson, Jaewoo Jung, and John E. Robinson III. "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations". In: *16th AIAA Aviation Technology, Integration, and Operations Conference.* 2016. DOI: 10.2514/6.2016-3292.

[7] FAA. *Urban Air Mobility Concept of Operations.* Tech. rep. Version 2.0. Federal Aviation Administration, 2023. URL: https://www.faa.gov/sites/faa.gov/files/Urban%20Air%20Mobility%20%28UAM%29%20Concept%20of%20Operations%202.0_0.pdf.

[8] Single European Sky ATM Research 3 Joint Undertaking. *European drones outlook study: unlocking the value for Europe.* Publications Office, 2017. DOI: 10.2829/085259.

[9] Malik Doole, Joost Ellerbroek, and Jacco Hoekstra. "Estimation of traffic density from drone-based delivery in very low level urban airspace". en. In: *Journal of Air Transport Management* 88 (2020), p. 101862. ISSN: 0969-6997. DOI: 10.1016/j.jairtraman.2020.101862. (Visited on 12/23/2020).

[10] Government of the Netherlands. *GoDrone Interactive Map.* https://map.godrone.nl/. Accessed: 2024-08-20.

[11] Emmanuel Sunil, Joost Ellerbroek, Jacco M. Hoekstra, and Jerom Maas. "Three-dimensional conflict count models for unstructured and layered airspace designs". In: *Transportation Research Part C: Emerging Technologies* 95 (2018), pp. 295–319. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2018.05.031. URL: https://www.sciencedirect.com/science/article/pii/S0968090X1830771X.

[12] Andres Morfin Veytia et al. "Metropolis II: Benefits of Centralised Separation Management in High-Density Urban Airspace". In: *12th SESAR Innovation Days.* 2022.

[13] Suda Bharadwaj, Steven Carr, Natasha Neogi, and Ufuk Topcu. "Decentralized Control Synthesis for Air Traffic Management in Urban Air Mobility". In: *IEEE Transactions on Control of Network Systems* 8.2 (2021), pp. 598–608. DOI: 10.1109/TCNS.2021.3059847.

[14] C. Badea, A. Morfin Veytia, M.J. Ribeiro, M. Doole, J. Ellerbroek, and J. Hoekstra. "Limitations of Conflict Prevention and Resolution in Constrained Very Low-Level Urban Airspace". In: *11th SESAR Innovation Days.* 2021.

[15] Malik Doole, Joost Ellerbroek, and Jacco M. Hoekstra. "Investigation of Merge Assist Policies to Improve Safety of Drone Traffic in a Constrained Urban Airspace". In: *Aerospace* 9.3 (2022). ISSN: 2226-4310. DOI: 10.3390/aerospace9030120. URL: https://www.mdpi.com/2226-4310/9/3/120.

[16] Daniel Delahaye and Stéphane Puechmorel. "Air Traffic Complexity: Towards an Intrinsic Metric". In: *Proceeding of the 3rd USA/Europe Air Traffic Management R and D Seminar.* Unknown, Unknown Region, 2000. URL: https://enac.hal.science/hal-01205255.

[17] Zhengyi Wang, Daniel Delahaye, Jean-Loup Farges, and Sameer Alam. "Complexity optimal air traffic assignment in multi-layer transport network for Urban Air Mobility operations". In: *Transportation Research Part C: Emerging Technologies* 142 (2022), p. 103776. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2022.103776. URL: https://www.sciencedirect.com/science/article/pii/S0968090X22002054.

[18] Andrija Vidosavljevic, Daniel Delahaye, Emmanuel Sunil, Frank Bussink, and Jacco Hoekstra. "Complexity Analysis of the Concepts of Urban Airspace Design for METROPOLIS Project". In: *EIWAC 2015, 4th ENRI International Workshop on ATM/CNS.* ENRI. Tokyo, Japan, Nov. 2015. URL: https://enac.hal.science/hal-01234078.

[19] Karl Bilimoria and Hilda Lee. "Analysis of Aircraft Clusters to Measure Sector-Independent Airspace Congestion". In: *AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences.* Arlington, Virginia, 2005.

[20] Niki Patrinopoulou, Ioannis Daramouskas, Calin Andrei Badea, Andres Morfin Veytia, Vaios Lappas, Joost Ellerbroek, Jacco Hoekstra, and Vassilios Kostopoulos. "Dynamic Capacity Management for Air Traffic Operations in High Density Constrained Urban Airspace". In: *Drones* 7.6 (2023). ISSN: 2504-446X. DOI: 10.3390/drones7060395. URL: https://www.mdpi.com/2504-446X/7/6/395.

[21] Michiel de Bok and Lóri Tavasszy. "An empirical agent-based simulation system for urban goods transport (MASS-GT)". In: *Procedia Computer Science* 130 (2018). The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops, pp. 126–133. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.04.021. URL: https://www.sciencedirect.com/science/article/pii/S1877050918303715.

[22] Joe H. Ward. "Hierarchical Grouping to Optimize an Objective Function". In: *Journal of the American Statistical Association* 58.301 (1963), pp. 236–244. ISSN: 01621459. URL: http://www.jstor.org/stable/2282967 (visited on 11/28/2023).

[23] Boris Mirkin. *Hierarchical Clustering*. en. Ed. by Boris Mirkin. Undergraduate Topics in Computer Science. London: Springer, 2011, pp. 283–313. ISBN: 978-0-85729-287-2. DOI: 10.1007/978-0-85729-287-2_7. URL: https://doi.org/10.1007/978-0-85729-287-2_7 (visited on 11/28/2023).

[24] Andres Morfin Veytia. *Supporting data and code for:* 2024. DOI: 10.4121/54825f14-8743-447d-8346-3afa46d319d6.v1. URL: https://doi.org/10.4121/54825f14-8743-447d-8346-3afa46d319d6.v1.

[25] Steven R. Lay 1944-. "Convex sets and their applications". English. In: Pure and applied mathematics (John Wiley & Sons). New York: Wiley, 1982, p. 48. ISBN: 0-471-09584-2 978-0-471-09584-2.

[26] Carolyn Reinhart. "The distance matrix and its variants for graphs and digraphs". Available at https://dr.lib.iastate.edu/handle/20.500.12876/RwyqEbkw. PhD thesis. Ames, Iowa: Iowa State University, 2021.

[27] Malik Doole, Joost Ellerbroek, Victor L. Knoop, and Jacco Hoekstra. "Constrained Urban Airspace Design for Large-Scale Drone-Based Delivery Traffic". en. In: *Aerospace* 8.2 (2021). DOI: 10.3390/aerospace8020038.

[28] Edsger W Dijkstra. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[29] Gianluca Bianchin and Fabio Pasqualetti. "Routing Apps May Cause Oscillatory Congestions in Traffic Networks". In: Dec. 2020, pp. 253–260. DOI: 10.1109/CDC42340.2020.9303866.

[30] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945). Publisher: [International Biometric Society, Wiley], pp. 80–83. ISSN: 00994987. DOI: 10.2307/3001968. URL: http://www.jstor.org/stable/3001968 (visited on 08/14/2024).

[31] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. 2017.

[32] Geoff Boeing. "OSMNX: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks". In: *Computers Environment and Urban Systems* 65 (2017), pp. 126–139. DOI: 10.1016/j.compenvurbsys.2017.05.004.

[33] Pratyush Tripathy, Pooja Rao, Krishnachandran Balakrishnan, and Teja Malladi. "An open-source tool to extract natural continuity and hierarchy of urban street networks". In: *Environment and Planning B: Urban Analytics and City Science* (2020). DOI: 10.1177/2399808320967680.

[34] J. Hoekstra, J. Ellerbroek, E. Sunil, and J. B. Maas. "Geovectoring: Reducing Traffic Complexity to Increase the Capacity of UAV airspace". en. In: *2018 International Conference on Research in Air Transportation* (2018). (Visited on 12/02/2020).

[35] Michiel de Bok, Lóránt Tavasszy, Ioanna Kourounioti, Sebastiaan Thoen, Larissa Eggers, Victor Mayland Nielsen, and Jos Streng. "Simulation of the Impacts of a Zero-Emission Zone on Freight Delivery Patterns in Rotterdam". In: *Transportation Research Record* 2675.10 (2021), pp. 776–785. DOI: 10.1177/03611981211012694. eprint: https://doi.org/10.1177/03611981211012694. URL: https://doi.org/10.1177/03611981211012694.

[36] Călin Andrei Badea, Andres Morfin Veytia, Niki Patrinopoulou, Ioannis Daramouskas, Joost Ellerbroek, Vaios Lappas, Vassilios Kostopoulos, and Jacco Hoekstra. "Unifying Tactical Conflict Prevention, Detection, and Resolution Methods in Non-Orthogonal Constrained Urban Airspace". In: *Aerospace* 10.5 (2023). ISSN: 2226-4310. DOI: 10.3390/aerospace10050423. URL: https://www.mdpi.com/2226-4310/10/5/423.

[37] International Civil Aviation Organisation. *Annex 10 - Aeronautical Telecommunications - Volume I - Radio Navigational Aids*. 7th ed. ICAO, 2018. ISBN: 978-92-9265-233-3.

[38] DJI. *DJI Matrice 600 Pro Support Page*. Accessed 2024. URL: https://www.dji.com/nl/support/product/matrice600-pro (visited on 05/14/2024).

[39] Jacco Hoekstra and Joost Ellerbroek. "BlueSky ATC Simulator Project: an Open Data and Open Source Approach". In: *International Conference for Research on Air Transportation*. 2016.

[40] Artur Gramacki. *Nonparametric kernel density estimation and its computational aspects*. English. Studies in big data; v. 37; volume 37. Cham, Switzerland: Springer, 2018. ISBN: 978-3-319-71688-6 3-319-71688-3. DOI: 10.1007/978-3-319-71688-6. URL: https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1668756.

[41] TUDelft-CNS-ATM. *BlueSky: The open source air traffic simulator*. https://github.com/TUDelft-CNS-ATM/bluesky, Accessed 18-Feb-2022. URL: https://github.com/andubadea/bluesky/tree/M2-sensitivity.