4D Trajectory Conflict Detection and Resolution Using Decision Tree Pruning Method

Lucas B. Monteiro Vitor F. Ribeiro Cristiano P. Garcia Geraldo P. Rocha Filho and Li Weigang
Senior Member, IEEE

***Abstract*—The aviation community develops Trajectory Based Operations (TBO) as an advancement in Air Traffic Management (ATM). There is still the need for an efficient scheme to present the trajectories, manage their associated data, and further detect and resolve the conflicts (CD&R) that should eventually occur. In this research, we develop a CD&R framework for managing predicted 4-Dimensional Trajectory (4DT). Using Not Only SQL (NoSQL) database (Cassandra and MongoDB), the 4D trajectories of related routes are presented, and the possible conflicts are detected using the strategy of Computing in NoSQL Database. Compared with other conflict detection algorithms, usually by the pairwise method with $O(n^2)$ at least, the proposed Decision Tree Pruning Method (DTPM) effectively treats massive data sets. The 4DT data are collected by Trajectory Predictor (TP) concerning 58% of the whole Brazilian air traffic. The comparison results between Cassandra and MongoDB from the case studies show the effectiveness of the proposed methods for conflict detection. In addition, we prove that the conflict resolution approach is viable for application in real scenarios, finding near-optimal solutions for the conflicts identified by the framework. Finally, we also demonstrated the development of sustainable artificial intelligence in intelligent air transportation to improve safety in air traffic management.**

***Index Terms*—4-Dimensional Trajectory, Conflict Detection and Resolution, Decision Tree Pruning Method, Not Only SQL.**

I. Introduction

An essential requirement for efficient Air Traffic Management (ATM) is the quality of available information, which must be up-to-date, highly accurate, and reliable. This enables the user to make the right decisions at the right time [1], thus supporting scenario prediction, resource allocation, and trajectory management.

Trajectory-based operations (TBO) is a new technology that defines strategic long-term conflict resolution trajectories that combine security and efficiency [2], as illustrated in Figure 1. Different stakeholders on the ground and in the air must share a standard view of the aircraft trajectories [3]. This task is made possible by the implementation of Four-dimensional (4D) navigation, which is a method that adds the time factor to spatial constraints, thus favoring the accurate prediction of the aircraft's position at a given time window [4]. As a particular computation language to support ATM, Aircraft Intent Description Language (AIDL) has been developed for this purpose [3].

Lucas B. Monteiro
Vitor F. Ribeiro
Cristiano P. Garcia
Geraldo P. Rocha Filho
Li Weigang

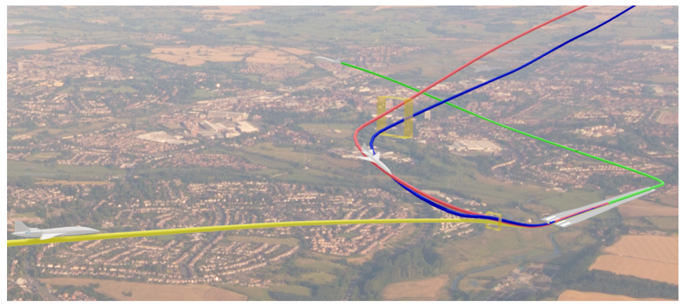


Fig. 1. Illustration of Trajectory-based operations. The figure shows three trajectories of aircraft approaching for landing at the airport and one takeoff trajectory. The yellow squares represent waypoints on an instrument approach chart (IAC).

Four-dimensional Trajectories (4DT) and Performance-based navigation (PBN) have become good subjects for research since airspace authorities worldwide established directives to implement these technologies. The scope of this work is the development of an intelligent system, which should support the operation of controllers to maintain a safe and efficient air transportation environment. Our research aims at developing a framework for Conflict Detection and Resolution (CD&R), which uses NoSQL databases and the Decision Tree Pruning Method to manage the 4-D-based trajectories on selected routes. The final product is a tool for CD&R among 4D trajectories. Our framework is designed to comply with the SWIM paradigm (System-Wide Information Management) and accesses an AIDL trajectory prediction service.

In the implementation of TBO considering 4D Navigation management increases the challenge for CD&R. Although this task is a traditional theme in ATM [5], CD&R can be divided into two main scenarios: without TBO and with TBO. The general procedure of CD&R can be described in two steps. The first step (1) is to predict and present the 4D Trajectory data, which can be estimated from a trajectory predictor. The second step (2) is to detect the conflicts. If there is any conflict, the procedure is forwarded to the third step (3) for the conflict resolution. As the result of step (3), the adjusted 4D trajectory data is back-propagated to step (1) until there is no conflict. Then, the procedure is forwarded to the fourth step (4) to output the Conflict-free 4D Trajectory data.

The following aspects pose a significant challenge in CD&R with the implementation of TBO: 1) highest safety requirement for ATM [6]; 2) massive data volume generated by regional and global traffic in 4D Navigation [7]; 3) uncertainty of human and environmental factors such as weather and temperature [8]; 4) repeated computation and management to achieve conflict-free scenarios [9]. There are some attempts, including the usage of Graphics Processing Units (GPU) to improve the computation performance [10], but this limitation remains unresolved as the statement of the problem in 1997 [11]. Notwithstanding, the critical problem is high computation complexity for Conflict Detection, for proximity $O(n^2)$, where n is the number of aircraft at a specific time, and it is necessary to check whether every aircraft are properly separated from each other.

As the main contribution of this research, a new framework of CD&R for 4D Navigation management is developed using a particular scheme of a Not Only SQL (NoSQL) database. Alongside with NoSQL, compared with other CD&R algorithms (usually using $O(n^2)$ pairwise method), the proposed algorithm uses the decision tree pruning method (DTPM), which makes it possible to operate a large amount of data to reduce the computational cost economically. In addition, the framework puts forward two conflict resolution methods, which benefit from the proposed conflict detection model and find the best solutions within a satisfactory time. The research shows that the development of artificial intelligence promotes the sustainable development of intelligent air transportation, thus effectively improving air traffic management.

The remaining of this paper is organized as follows: the state of the art in CD&R is discussed in Section II. In Section III, the proposed methods for CD&R are described. Section IV shows the evaluation of the proposed approach. The conclusion of the experiment is reported in Section V.

II. Related Work

The need of new procedures in Air Traffic Flow Management (ATFM) tasks, especially in conflict detection and resolution regarding to 4D trajectories, automatically instigates the enhancement of the existing technologies. However, despite the rapid evolution of equipment and software, the computational requirements for storing and evaluating the available data are becoming increasingly challenging [12].

Ruiz presents a method based on causal modeling for collaborative conflict resolution [13]. A set of user-preferred trajectories is stored in the first moment, and potential conflicts are discovered. Locally optimal trajectories are generated for each pair of trajectories and conflicts by considering different types of maneuvers to solve conflicts. Conflict detection is evaluated iteratively in this alternative set. Causal exploration with constraint propagation evaluates branches of the reachability tree of each trajectory and propagates the constraints through the feasible solution set, which reduces the search space.

Some solutions develop sophisticated methods by reshaping the airspace and applying proper algorithms suitable to the modeled environment. Ayhan et al. [14] model the space as a grid of cubes whose centroids are 4D trajectory waypoints. Conflicts are detected when two or more cubes overlap, and the resolution is based on a Hidden Markov Model (HMM) that learns from the historical trajectories and their correlation with weather parameters. An adapted Viterbi algorithm rearranges the cubes to prescribe new 4D trajectories. Although their solution is very useful for strategic deconfliction, the quadratic complexity inherent to the pairwise CD&R can be improved.

Tang [11] presented an up-to-date review of the literature, comparing different short, medium, and long-term approaches, and designing a framework detached from the CD&R. Sandamali et al. [15] proposed treatment of an approach to ATFM focused on the associated uncertainties, through a probabilistic model based on chance. The main advantage of the proposed approach is the ability to process volumes of data online, with reduced search space.

Trajectory-Based Operations (TBO) were studied by Sabatini et al. [16], and Cai et al. [17], among others. Both regard 4D trajectories as airspace resources that must be distributed among the aircraft efficiently and safely in terms of costs and conflict avoidance. Sabatini's approach models a negotiation scenario with multiple trajectory options for every aircraft where all 4DT intents are pairwise-compared using a rule-based model, while Cai's approach use interpolation on the trajectory waypoints to detect the conflicts and later a genetic algorithm is applied to solve the conflicts regarding the fairness on cost distribution.

Rosenow et. al. [18] established an approach to flight path optimization with a focus on flight sustainability. The objective is to identify how climatic uncertainties that demand trajectory changes impact the sustainability of flights. Because the emphasis is on optimizing the trajectory and influence of the flight, the consequences of continuous trajectories in the whole air space were not observed. This problem will be addressed in this work by Decision Tree Pruning, and the alternative trajectories of each flight will be considered in the modeling, which will be verified in the next sections.

In the field of decision trees, Malakis et. al. [19] presents a study focused on the accurate classification of air traffic scenarios, and on how such classification can help to better understand how flight controllers respond to the complexity of a traffic situation. A perceived difference for this work is in the focus given to the investigation of the behavior of controllers, and not the detection and resolution of conflicts. That is, they are different mechanisms of assistance to the activity of air traffic control.

Liu et. al. [20] propose a novel approach to predict the actual aircraft 4D trajectories, using high-dimension meteorological features and last filed flight plans. A framework was developed composed of a matching algorithm, a deep generative model, a training framework, and an inference framework. The actual flight state of latitude, longitude, altitude, latitude speed and longitude speed was modeled as conditional Gaussian mixtures, and the parameters will be learned from the proposed depth generation model. It is a research with a different focus from the one presented in this work (prediction of trajectories vs. CD&R), however they are complementary and a new version of the present framework considering this new modeling for prediction of conflicts can be studied.

Wang et. al. [21] presents a generic modeling for three-dimensional CD&R, which separates resolution and conflict detection, allowing the comparison of different models. Compared with the proposed model, the main differences lies in the separation of conflict detection and problem solving. We deal with this problem in a single step, so as to reduce the overhead and eliminate the need to completely reprocess the trajectories when a conflict is detected, as well as the time required for each cycle of the algorithm. In the proposed algorithm, the algorithm focuses on proposing solutions within up to 30 seconds, thus reducing the impact of the final change/uncertainty in the trajectories.

Finally, Tang [22] introduced a recent review of CD&R literature, divided these studies into 3 different categories: long-term, medium-term and short-term, and classified them according to common features put forward, such as conflict detection and resolution, initiation of new conflicts, cooperation between aircraft, maneuver of execution, global or paired execution, etc. The present research, which can be framed in the short-term group, gathers characteristics that are desirable in each of the categories, being the only one from the presented list to be concerned with induced conflicts among the short-term approaches.

III. Proposed Solution

The predicted trajectories are evaluated for conflict detection. Thus, resolving conflict means adjusting the predefined flight plans to update the predicted trajectories. The complete workflow for the proposed conflict detection and resolution is shown in Figure 2. The main parts of the models are explained in the following subsections.

A. Trajectory Prediction

A raw trajectory is a sequence of two-dimensional points that form a connected set of segments. The set of waypoints $P = (p_1, ...p_n)$ is able to describe a trajectory in a 2D plane, but it must be refined to accomodate further control variables and a chronological constraint in order to describe a complete 4D trajectory [24]. In other words, we need a set $W = w_1, ..., w_n$ provided that $w_i = (x_i, y_i, z_i, t_i) \in \mathbb{R}^4$ is a 4D-trajectory point where x_i, y_i, z_i are spacial dimensions and t_i is time.

Trajectory computation must regard not only the predefined set of waypoints informed in the flight plan (origin, destination, and some fix points in between) but also the parameters that affect the flight profile, such as takeoff mass and velocity. Weather and aircraft performance models form the flight script and are necessary input to any trajectory predictor system [25]. The product of the trajectory computation process is called Aircraft Intent, which informs precisely how the aircraft should comply with the flight plan. As a result, unambiguous description of the intended flight path is delivered.

The continuous communication of complete and up-to-date information between the aircraft and ground service providers enable proper decision making processes [26]. In the 4D navigation context, the main information is the conflict-free trajectories calculated by the ground operators that must be issued to the airborne aircraft. This is only possible when there is common awareness of the air traffic scenario, which means unequivocal understanding of the flight intentions. The Aircraft Intent Description Language (AIDL) is used for this purpose [27]. This language was originally proposed by Vilaplana et al [4].

The framework built under the AIDL concept, see Figure 2, is used as a service for the trajectory computation in the tool we propose in this work. The trajectory prediction used in this framework receives the aircraft parameters and other information about the flight intention at a given moment, and returns the calculated trajectory. This process is performed iteratively, and at each new cycle, the trajectory points calculated by the trajectory prediction are used in the conflict detection process.A detailed description of the motion equations and their results can be found in the work of Vilaplana et al [27].

An important feature of this framework is that the modeling not only considers the standard predicted trajectories but also considers a group of alternate trajectories. The whole trajectory set is calculated simultaneously, which allows a conflict between predicted trajectories to be identified and the aircraft are grouped in clusters of conflicts. Therefore, it is possible to seek a solution to combine the predicted trajectories and alternate trajectories to ensure that the decision to perform a maneuver (the selection of an alternate trajectory) will not generate a new conflict. Furthermore, as the alternate trajectory has also been calculated, the possible conflicts that this new trajectory could cause can be evaluated by the framework at the moment of selection.

B. 4D Trajectory Storage Proposal

We designed the proposed NoSQL databases to be compliant to the SWIM paradigm since they can be used to massive trajectory storage, weather models, and aircraft data in any detail level. NoSQL refers to an increasing group of nonrelational data management systems where databases are not built primarily on tables and generally do not use SQL for data manipulation. When the nature of data doesn't require a relational model, NoSQL database management system is very useful in dealing with large amounts of data [28]. Although there is no advantage for a specific type of NoSQL database, compared with the traditional SQL database, the performance advantage for NoSQL is obvious [29]. Therefore, we conducted preliminary tests on other types of NoSQL databases, among which Cassandra and MongoDB performed best among the evaluated technologies. In order to implement the functionality of conflict detection in 4D trajectories, one should define the data type to be retrieved. Relationships among the entities are not possible through foreign keys. Thus every distinct relationship depends on the implementation of distinct entities: column families for Cassandra and collections for MongoDB, which are important examples of these kinds of databases. This might result in data replication, but in NoSQL paradigm, it is actually regarded as a feature instead of a drawback.

Probable queries desired in this implementation should be: Flight plans of an aircraft, Trajectory of an aircraft, Points in a trajectory, Points occupied at time t, and Trajectory conflicts.

The Trajectory Predictor application is able to calculate trajectories described in AIDL and generate a KML file as output. This file uses XML syntax to represent each sample

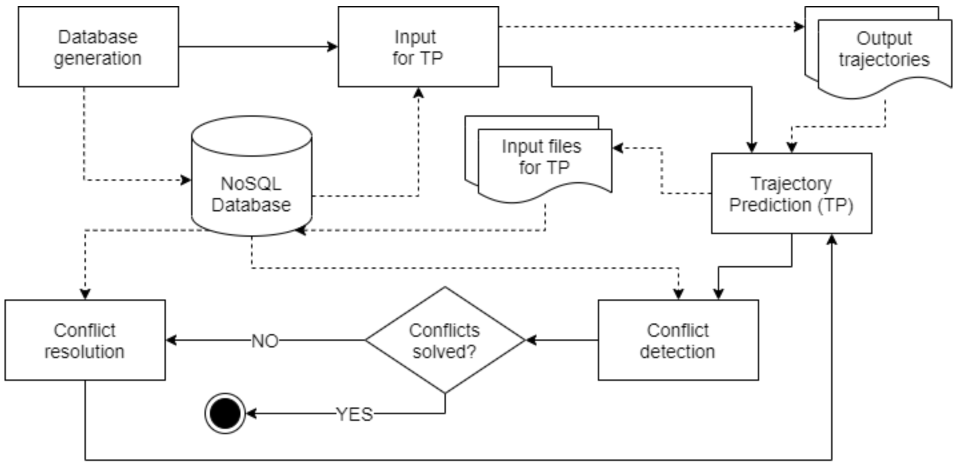


Fig. 2. Sequence diagram for conflict detection and resolution [23].

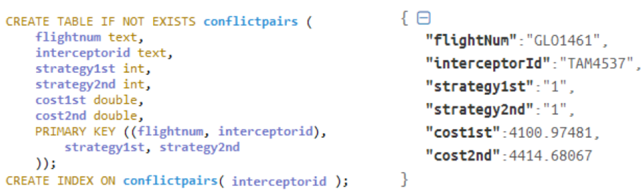


Fig. 3. Example of Cassandra and MongoDB data structure.

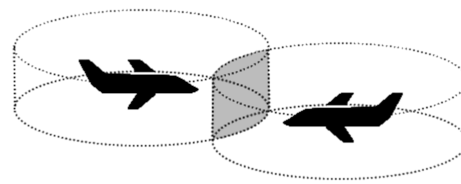


Fig. 4. Violation of exclusive zones between two aircraft.

point in a 4D trajectory. Thus, the sample point attributes (x, y, h, t) of a predicted trajectory can be used as input to any conflict detection tool. In the proposed approach, the partition key is formed by the flight number and the trajectory strategy. A clustering order on the column named *time* is applied so that the trajectory waypoints are ordered by their timestamp in ascending order. The entities are similarly designed both in Cassandra and MongoDB databases and can be queried to bring the whole set of (ordered) waypoints that form some aircraft's trajectory.

Our system implements a Cassandra and a MongoDB databases to store data to facilitate conflict detection. Since the conflict detection algorithm needs to manipulate a large amount of data, each information domain was modeled as a distinct column family or collection of documents for better performance to read and write data, depending on the subject database. Each modeled entity represents the information corresponding to a record in a relational table. Figure 3 presents an example of the final data structure in both Cassandra and MongoDB modeling. We also published the dataset in GitHub, see link: https://github.com/lucasbmonteiro/translab-cdr.

C. Conflict Detection

A conflict involving two or more airborne aircraft is the scenario where the minimum separation among them is compromised at one or more points in time. Both horizontal and vertical minimum distances must be ensured during the whole execution of the flights, which are defined respectively as 1,000 feet at flight levels below/above 29,000 feet, and five nautical miles. Therefore, every exclusive zone around an aircraft is a cylinder centered on it. The time constraint is added to the 4D trajectories, so the conflict is detected if this exclusive spatial zone is violated within a specific time window. Figure 4 illustrates a situation where two aircraft are in conflict.

Then, it is possible to identify the logical relation that defines the existence of a conflict c between aircraft A_i and A_j at an instant t, according to

$$c^{A_iA_j}(t) \leftrightarrow ({d_h}^{A_iA_j}(t) < S_h) \land ({d_v}^{A_iA_j}(t) < S_v) \quad (1)$$

where ${d_h}^{A_iA_j}(t)$ is the horizontal distance between A_i and A_j at instant t, ${d_v}^{A_iA_j}(t)$ is the vertical distance between A_i and A_j at instant t and S_h and S_v are, respectively, the minimum required horizontal and vertical separation.

The horizontal distance between points w_i and w_j is given by the Haversine formula, given by Equation 2. The equation computes the geodesic distance between the projections of 2D points to the Earth's sphere surface, where latitude and longitude are expressed in radians, and R is the Earth's radius.

$$\begin{gathered}\Delta\phi = |lat_i - lat_j| \\ \Delta\lambda = |long_i - long_j| \\ a = sin^2\Big(\frac{\Delta\phi}{2}\Big) + cos(lat_i) \cdot cos(lat_j) \cdot sin^2\Big(\frac{\Delta\lambda}{2}\Big) \\ c = 2 \cdot arctan2\Big(\sqrt{a}, \sqrt{1-a}\Big) \\ d_h^{w_i,w_j} = R.c\end{gathered} \quad (2)$$

The conflict detection algorithm implements the separation restrictions between the aircraft described in the Equation 1: at a specific time interval, two points must be at a minimum distance both vertically and horizontally. If all the conditions expressed by the equation are satisfied, the points compared

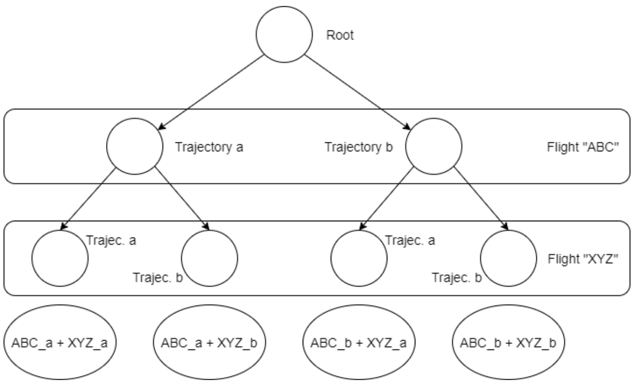


Fig. 5. Example of trajectory combination tree model.

conflict; otherwise, if at least one of the conditions fails, there is no conflict between the analyzed points.

The proposed approach allows the points to be analyzed in pairs, guaranteeing greater efficiency in the reduction of search space: as the points are ordered by its altitude, if there is a vertical separation between the first and second points, it is possible to infer that there is no conflict between the first and third points and the following points. In other words, no further verification involving the first point is required. This approach enables, in the best case, the identification of all conflicts in linear time. If there is a vertical conflict between two successive points, the algorithm then verifies if there is a minimum horizontal separation between both, registering a positive conflict case. If there is no conflict, the algorithm checks whether there is a violation of the minimum vertical separation between the first and third points (worst case and average case). From the point that there is no vertical separation violation, the first point is decayed, that is, it does not need to be parsed again.

D. Conflict Resolution

In the conflict detection stage of the framework, conflict clusters were formed, each cluster containing the totality of aircraft involved in some conflict and their respective trajectories, default and alternatives. The challenge becomes to identify, for each cluster, an optimal combination of individual trajectories that not only represent the non-occurrence of conflict, but also the most satisfactory combination, based on some previously chosen policy (flight cost, priorities, impact on airspace, etc.).

In conflict resolution modeling, the possible states, that is, the combinations of all trajectories of all aircraft, were organized in the form of the tree data structure, so that each level of the tree corresponds to an aircraft, each node of a level corresponds to a trajectory of that aircraft and a path between the root node and a leaf node represents the choice of a combination of trajectories. This modeling are represented in Figure 5.

Each node of the tree was assigned a score (parameter) that represents how suitable that choice is, that is, nodes that represent a higher score indicate a worse choice, while nodes that represent a better choice have a lower score. In this way, nodes that indicate trajectories involved in conflicts have a maximum score, in order to discourage the choice of any combination that contains that node (paths that reach a leaf node through this node).

The suggested modeling makes it possible to identify an optimal combination of trajectories, without conflicts, using classical search algorithms. Two approaches based on this class of algorithms were implemented: i) brute force, based on the depth-first search (DFS) algorithm; and ii) PRUNE, based on alpha-beta.

DFS Approach

The approach based on DFS consists of the complete assembly of the tree (all possible states), and the search by means of brute force, that is, all nodes are visited. It is an advantageous algorithm in the sense that it requires linear space in relation to the search depth [30], but it has the disadvantage of becoming unfeasible as the size of the tree increases (in number of flights and trajectories).

In the IV-B section, it is possible to verify that such an approach presents excellent results in smaller scenarios, that is, fewer aircraft and/or fewer trajectories per aircraft. It is likely that the DFS meets most of the current air traffic needs, but considering the prospect of a continuous increase in air traffic demand [31], it was necessary to develop a second model that allows the identification of optimal trajectories in complex scenarios.

PRUNING Approach

In the PRUNING approach, inspired by the classic Alpha-Beta algorithm, the assembly and tree search try to avoid that all possible nodes are investigated. This feature is obtained with the computation, simultaneously with the assembly of the tree, of the accumulated score up to the study node, eliminating the need to add new nodes if a cost worse than the minimum achieved so far is already identified.

The elimination of known worse nodes/paths allows the algorithm to reduce the search space, increasing the ability to analyze larger trees considering the same computational resources used in DFS. In the IV-B section it will be possible to identify this reduction in the search space and, consequently, the analysis of much more complex scenarios, probably sufficient for the current and future situation of global air traffic.

E. Space Search

A problem space consists of a set of states of the problem, and a set of operators that change the state. For example, in the Eight Puzzle, the states are the diferent possible permutations of the tiles, and the operators slide a tile into the blank position. A problem instance is a problem space together with an initial state and a goal state. A problem-space graph is often used to represent a problem space. The states of the space are represented by nodes of the graph, and the operations by edges between nodes. Edges may be undirected or directed, depending on whether their corresponding operators are reversible or not. The task in a single-agent path-finding problem is to find a path in the graph from the initial node to a goal node [30].

Storing data about these possible states requires enormous computing power. for example, if we consider 5 aircraft in the same cluster, with 40 trajectories for each aircraft (1 default and 39 alternatives), to identify the optimal solution, it will be necessary to evaluate 10^8 possible combinations between each of the trajectories of each flight. The number of possible states that can be analyzed increases exponentially and can also be expressed by Equation 3:

$$E(Q_a, Q_t) = Q_t^{Q_a} \tag{3}$$

Where, Q_a indicates the number of aircraft and Q_t the number of trajectories per aircraft, default and alternatives. Equation 3 can be used to calculate the number of possible states (search space). For complex configurations, it is necessary to use computing technology to evaluate as many situations as possible. It can also show effectiveness in the big data environment through the suggested method.

IV. Case Study

This section presents the case study that uses real data from the Brazilian air traffic scenario to evaluate the applicability of the proposed approach using Cassandra and MongoDB.

A. Simulation Scenario

Two different databases were used to analyze the proposed framework: one based on real data, focusing on the performance of the conflict detection component; and another with randomly generated data, for evaluation of the conflict resolution component.

The simulation scenario for conflict detection is composed by a selection of nine airports in Brazil. We chose to use information from local air traffic, despite Brazil being at the forefront of the world in terms of operational safety in civil aviation [32], because the growth of air traffic demand requires the continuous development of new methods and operation modes, which can ensure the safety of air transportation even when the scale increases. The country is the second in the world in number of airports, and the selected ones serve the majority of the Brazilian population. Combined, they were responsible for 58% of the national air traffic in 2018 [33]. Therefore they are a proper representation of the Brazilian air traffic scenario for this simulation.

The repetitive flight database was collected from the free website of the Brazilian Air Navigation Management Center (CGNA), which contains a list of flight plans with their respective validity periods. Among the valid flights, those with validity until June 2019 were selected. This selection did not consider the frequency and periodicity of such flights in general. Among the selected plans, 1,221 subject flights are filtered for simulation, as long as these flights have origin and destination airports in the selected airport set.

Once the database is fully populated, all flight plans (RPL) are fetched and submitted to the engine that generates the input files for TP. Then, the batch script executes the trajectory prediction for each flight. The client application accesses the output trajectories and inserts every sampled waypoint into the database. After running the trajectory prediction module, it was possible to have a comprehensive overview of the scenario occupation.

In a second simulation scenario, the flight plans were replicated with updated flight levels and Mach speeds to create different strategies, equivalent to alternate trajectories. In total, the distinct parameter combinations yield five different alternative trajectories to be selected by the aircraft subject to decision making.

Simulated scenarios with randomly generated data were used to evaluate the proposal for conflict resolution, allowing greater control and complexity of the tests, since in the executions with real data, the identified conflict clusters were composed, mostly, by few combinations of aircraft. In this way, fictitious clusters with few to dozens of aircraft in conflict were considered, and a number of trajectories in the order of up to hundreds of possible cases.

The tests were performed considering the same input parameters, both for the DFS embroidery and for the PRUNING approach. Scenarios with different numbers of aircraft in the same cluster were considered (3, 5, 7, 9, 12, 15, 18, 20, 25 and 30 aircraft), and, for each scenario, different amounts of trajectories considered for each aircraft, that is, the standard trajectory and as many alternative trajectories (3, 7, 10, 15, 20, 25, 30, 40 and 50).

Because the simulation data is used in the analysis of conflict resolution method, it was decided to define the processing time as the stop standard, that is, the number of flights and trajectories increases to a certain limit of execution time, which could make it infeasible in the real world. Therefore, only the results of the conflict resolution algorithm whose duration does not exceed 120 seconds were considered.

Additionally, for each combination of number of aircraft and number of trajectories per aircraft, successive tests were performed (30 repetitions), in order to ensure that the results obtained are statistically sustainable, totaling 2,700 executions.

B. Results

In this subsection, the results of tests carried out with the proposed framework will be presented, focusing mainly on the performance of the CD&R models, considering the large volume of processed trajectory data.

Conflict detection and Performance evaluation

The flight path of each aircraft is represented as a sequence of waypoints sampled by TP. A total of 160,022 waypoints were sampled by the trajectory predictor, provided that the database was filled with data corresponding to a whole day of operation.

Following the trajectory prediction, the conflict detection procedure is triggered, and the conflicts found are inserted into the database as well. The waypoint database is then queried by time window, and the points occupying the airspace in the given moment of time are returned. For simulation purposes, three conflict detection procedures were performed: (i) *Default conflicts* the conflicts naturally found if the flight plans are performed as originally defined by the RPL; (ii) *Alternate conflicts*, the conflicts found among all original trajectories plus the alternate trajectories; and (iii) *Time-window conflicts*, the conflicts are evaluated within a specific look-ahead time.

The simulations were performed with the following configurations: a HP Z220CMT BR Workstation equipped with

TABLE I

PERFORMANCE OF CONFLICT DETECTION AMONG DEFAULT TRAJECTORIES. THE NUMBER OF CONFLICTS IS ALWAYS THE SAME BECAUSE THE TESTS WERE RUN WITH THE SAME DATASET.

Threads	Waypoints	Conflicts	**Cassandra** Time (s)	**MongoDB** Time (S)
1	160,022	330	0.936	0.518
2	160,022	330	0.516	0.269
4	160,022	330	0.331	0.2
8	160,022	330	0.286	0.159
16	160,022	330	0.298	0.19
32	160,022	330	0.25	0.207

TABLE II

PERFORMANCE OF CONFLICT DETECTION AMONG ALL STRATEGIC TRAJECTORIES. THESE NUMBERS SHOW THAT THE RESULTS ARE CONSISTENT BY INCREASING THE NUMBER OF THREADS TO SPEED UP THE EXECUTION OF THE ALGORITHM, AND THE EXECUTION TIME CONTINUES TO BE SHORTENED WHEN 8 THREADS ARE ADDED.

Threads	Waypoints	Conflicts	**Cassandra** Time (s)	**MongoDB** Time (S)
1	833,072	8,466	1m12,025s	1m20,319s
2	833,072	8,466	0m38,126s	0m42,293s
4	833,072	8,466	0m23,075s	0m24,802s
8	833,072	8,466	0m15,798s	0m19,169s
16	833,072	8,466	0m15,498s	0m18,605s
32	833,072	8,466	0m16,216s	0m17,042s

a Intel® Xeon® CPU E3-1270 V2 3.50GHz, 32GB DDR3 SDRAM 800MHz and Microsoft Windows 7 Professional 64-bit as operating system. This situation allows the conflict detection process to be executed in parallel and in sequence, and the allocation of threads is managed programmatically. Table I presents the performance of conflict detection using the default trajectories only. The execution was performed using up to 2^n threads, where $0 \leq n \leq 5$. Similarly, the strategic approach was also evaluated. Table II show the performance of conflict detection using the combination of all possible trajectories to be performed by the aircraft, with different combinations of flight plan executions.

Figure 6 shows an example of a conflict between two flights. The trajectory represented by an orange path belongs to a flight departing from Galeão to Congonhas, whilst the trajectory represented by a teal line is performed by a flight departing from Santos Dumont to Congonhas. Although their distinct cruise levels (FL340 and FL300 respectively) grant the safe separation for most part of the flight, the loss of separation is detected when both aircraft are in the descent procedure.

It is essential to clarify that the proposed conflict prediction procedure comprises a whole day of operations. Thus every scheduled flight is evaluated. For each flight, four more strategies are provided, meaning that every aircraft has five different alternate trajectories to be performed in case of conflict. This justifies the large number of conflicts found: Firstly, the default flight plans insert 330 conflicts in the scenario throughout the day. The novel conflict detection methodology presented in this work also takes into consideration the probable conflicts that should appear in every possible combination of pre-

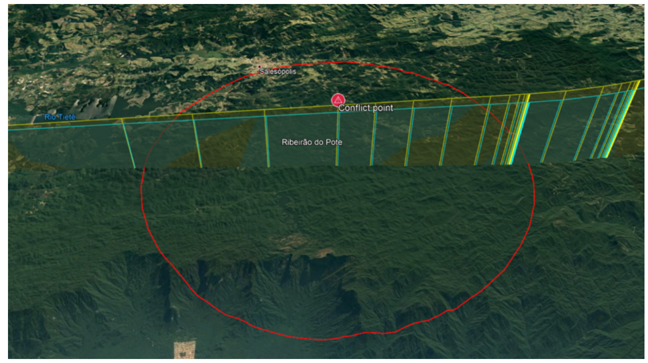


Fig. 6. A conflict detected between two aircraft.

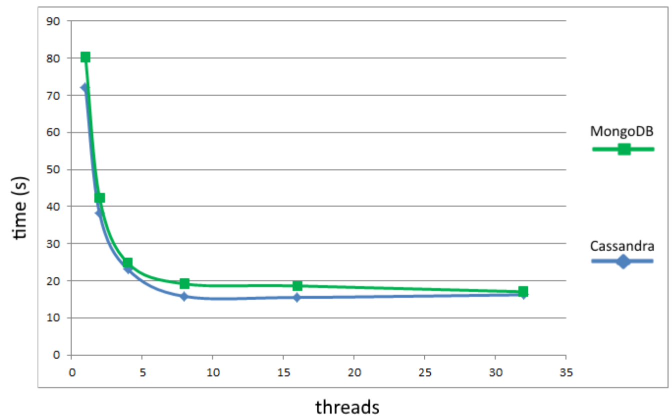


Fig. 7. Performance comparison with threaded speed-up.

selected alternate trajectories summing up to 6,105 trajectories and 833,072 waypoints.

Although this procedure completes the strategic conflict prediction in a whole day, further assessment was performed for sequential time windows comprising the evaluation of one hour from the current moment. This evaluation is important to demonstrate the efficiency of the algorithm in a dynamic scenario where conflicts should be detected on-demand, typically in a short-term tactical operation, to be implemented as part of a future work. Table III shows the performance obtained for conflict detection for default trajectories and alternate trajectories during odd time period. Both scenarios were performed in parallel execution with 16 threads.

The algorithm for conflict detection was verified to be far more efficient than the algorithms found in the state-of-art survey concerning the execution time. In fact, the NoSQL database method eliminates the need for data pre-processing for applications, so the data set provided is considered as a pruned search space. About adding threads to speed up the execution, we found the consistent results. As expected, Figure 7 shows that the execution time dropped consistently up to 8 threads due to the CPU core virtualization feature.

Conflict Resolution Analysis

Table IV shows that DFS algorithm can find the optimal solution within the predetermined time limit for as many as 10^8 possible combinations of search spaces. This limitation means that with the increase of the number of flights, the number of trajectories will decrease, which is a predictable

TABLE III
TIME-WINDOW CONFLICT DETECTION FOR DEFAULT AND ALTERNATE TRAJECTORIES.

	Default trajectories				**Strategic trajectories**			
Hour	Waypoints	Conflicts	**Cassandra** Time (s)	**MongoDB** Time (s)	Waypoints	Conflicts	**Cassandra** Time (s)	**MongoDB** Time (s)
1:00	4,764	2	0.012	0.007	24,316	96	0.296	0,228
3:00	1,018	0	0.002	0.002	5,133	11	0.096	0,074
5:00	0	0	0.002	0.002	0	0	0.002	0.001
7:00	405	0	0.002	0.001	2,158	0	0.002	0.002
9:00	6,086	8	0.015	0.016	32,484	179	0.282	0.307
11:00	10,220	22	0.013	0.012	53,166	563	1.505	1.421
13:00	11,135	30	0.021	0.008	58,029	711	1.277	1.3
15:00	9,530	24	0.012	0.01	50,101	673	0.952	0.929
17:00	8,613	14	0.017	0.024	44,495	401	0.496	0.521
19:00	9,112	16	0.017	0.008	47,306	410	0.484	0.52
21:00	10,692	29	0.02	0.011	55,789	768	0.626	0.544
23:00	9,320	22	0.033	0.046	48,065	550	0.39	0.297
Total	**160,022**	**330**	**0.372**	**0.278**	**833,072**	**8,466**	**13.772**	**13.951**

TABLE IV
TESTS RESULTS FOR DFS APPROACH.

Flight	Traject.	Execution Time (ms)	Possible = Verified Steps
3	3	0.33	10^1
3	20	2.43	10^3
3	50	22.30	10^5
9	3	7.83	10^4
9	7	8,737.60	10^7
12	3	145.27	10^5
15	3	3,858.57	10^7
18	3	106,181.87	10^8

TABLE V
TESTS RESULTS FOR PRUNING APPROACH.

Flight	Traject.	Execution Time (ms)	Possible Steps (p)	Verified Steps (v)	Search (v/p)
3	10	0.00	10^3	10^1	10^{-2}
3	50	0.43	10^5	10^2	10^{-3}
9	3	0.13	10^4	10^1	10^{-3}
9	7	0.63	10^7	10^2	10^{-5}
9	30	7.07	10^{13}	10^3	10^{-10}
9	50	15.20	10^{15}	10^4	10^{-11}
18	3	3.57	10^8	10^2	10^{-6}
18	10	58.07	10^{18}	10^3	10^{-15}
18	50	924.20	10^{30}	10^5	10^{-25}
30	7	3,939.13	10^{25}	10^3	10^{-22}
30	25	39,099.73	10^{41}	10^4	10^{-37}
30	30	47,622.43	10^{44}	10^5	10^{-39}

behavior. In a word, if the model considers as many as three trajectories (one default trajectory and two alternative trajectories) for each aircraft in a scene or clusters involving 18 aircraft in a conflict situation, it is possible to find the best combination of trajectories within the time limit, and find a solution within 1m46s on average, covering 100% search space of 10^8 states orders. At the other extreme, considering clusters with three conflicting aircraft, the algorithm can find the optimal solution considering 50 trajectories of each aircraft in only 22.3ms, covering 100% of a search space.

These figures show that the model becomes feasible in the case of fewer aircraft, which may be enough for the current air transportation situation. As an example, considering the assumption that 5 aircraft are in a conflict, the algorithm would be able to process as many as 40 alternative trajectories for each aircraft in about 18 seconds (average), which allows the aircraft to work comfortably in uncertain conditions.

Table V indicates that the PRUNING approach can handle even larger conflict clusters, with parameters high enough to guarantee coverage of the cases found in a real scenario. It was possible to solve, at one extreme, a cluster of conflicts with 30 aircraft and 30 trajectories for each aircraft, with a solution found, on average, in less than 50 seconds. If we consider that, in real scenarios, the clusters should not exceed 15 different aircraft, the model found the optimal solution, in an average time of 250.57 ms, processing 50 trajectories for each of the aircraft involved in the conflict (search space equivalent to 10^{25} For scenarios with fewer aircraft, which should be more

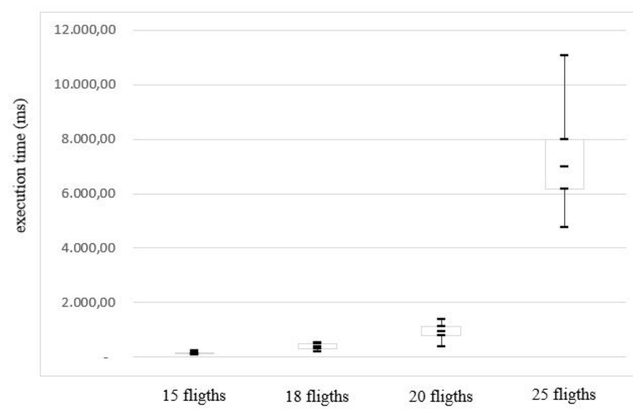


Fig. 8. Time execution distribution for PRUNING approach.

frequent, the solution was found in approximately 1 ms.

The stability of the model can be evaluated by continuously repeating the same scenario (the combination of flight times and each flight trajectory). Figure 8 shows the execution time distribution of different numbers of aircraft, focusing on the scenario with the largest number of trajectories. It is possible to identify, by the graph, that the most of the executions were close to the average performance, that is, average execution time, indicating that the proposed model is stable.

Finally, a conclusive analysis considers the comparison between the DFS and PRUNING method with respect to the real scenario, that is, the number of aircraft serving the current air traffic and the estimation of the trajectory of each aircraft. Figure 9 shows the surface corresponding to the distribution of the results obtained by DFS (a) and PRUNE (b). Taking into account the axis execution time, number of flights in a conflict group and the computed trajectories of each aircraft in the group, it proves the behavioral difference between these two methods. Considering that, in the real scenario, the algorithm should not take more than 120 seconds to present the solution, most of the surface corresponding to the DFS approach (a) is above the virtual plane of 120,000 milliseconds, while the surface corresponding to the PRUNE approach (b) lies entirely below the same virtual plane. Therefore, for all the test combinations shown in the figure, PRUNE managed to find the solution in a satisfactory time, while in the DFS method, this behavior was only verified in the simplest case (near the starting point, there are fewer aircraft and fewer trajectories per aircraft).

The results show that the proposed methods are feasible in the real world in terms of execution time. It can be seen in Table V, that, considering the possibility of executing the two methods in an acceptable time (less than 120 seconds), the PRUNING method always shows better results than DFS method. In the most complicated cases, considering 18 aircraft in conflict and 3 trajectories for each aircraft, compared with DFS method (106.2 seconds), PRUNING method is almost 30,000 times faster (3.57 milliseconds). Also in these cases, considering PRUNING method, it is possible to find a satisfactory solution to the conflicts in the cluster in less than 1 minute (47.6 seconds). These conflicts are more complicated than what should be observed in the actual scene (conflict involve 30 different aircraft, 30 alternative trajectories are computed for each aircraft, and a search space of 10^{44} possible states is generated). One possibility of adoption by the authorities is to integrate the model with the necessary tools for the inputs expected by the framework, comparing the results obtained by the application with the instructions issued by the ATC. In the tests, uncertain factors, such as noise and weather conditions, were not considered. However, by adding virtual obstacles or random disturbances to the trajectories, the model can adapt to this type of constraint. This kind of simulation will be considered in the next step of research.

V. Conclusion

The main contribution of this research is developing a new computational solution for CD&R in 4D trajectories that incorporates a trajectory predictor, decision tree pruning method, and databases specifically designed for big data. The developed application consumes information from several sources conveniently aggregated in NoSQL Cassandra and MongoDB databases. It invokes an external service for trajectory predictions in a successful attempt to adequate the ATM operations to the SWIM architecture paradigm.

Novel methods for 4D trajectory evaluation were presented by comparison between Cassandra and MongoDB. The proposed storage architecture is designed to comply with SWIM

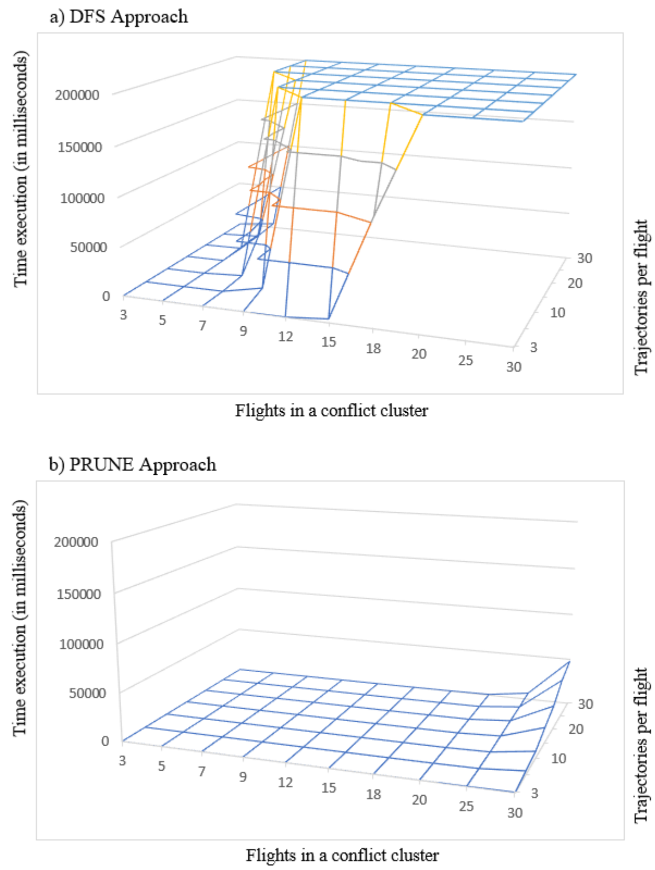


Fig. 9. Comparison between proposed approaches, DFS (a) and PRUNE (b).

paradigm, as this architecture can manage and provide information to support the decision-making process of air traffic stakeholders. However, this application does not mean to replace the air traffic controller (ATCO), but as a decision-making support tool for the human operator. This part of the Human-Machine integration in ATM will be further studied as a new direction for Intelligent Air Transportation.

As future work, other types of search algorithms can be studied in terms of performance and compared with the two methods introduced in this paper. Additionally, other NoSQL database technologies can be investigated as well, evaluating possible impacts on the presented modeling.

VI. Acknowledgements

This work has been partially supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under the grant number 309545/2021-8 and also by Boeing Research & Technology/Brazil. Geraldo P. Rocha Filho would like to thank the FAPESP (São Paulo Research Foundation) under grant #2021/06210-3.

References

[1] J. E. Dieudonne, H. L. Crane, S. R. Jones, C. J. Smith, S. A. Remillard, and G. Snead, “Neo (nextgen 4d tm) provided by swim’s surveillance soa (sdn asp for rnp 4d ops),” in *2007 Integrated Communications, Navigation and Surveillance Conference*, pp. 1–12, April 2007.

[2] A. Rodríguez-Sanz, D. Álvarez, F. Comendador, R. Valdés, J. Pérez-Castán, and M. Godoy, "Air traffic management based on 4d trajectories: A reliability analysis using multi-state systems theory," *Transportation Research Procedia*, vol. 33, pp. 355 – 362, 2018. XIII Conference on Transport Engineering, CIT2018.

[3] G. Frontera, J. Besada, A. Bernardos, E. Casado, and J. López-Leonés, "Formal intent-based trajectory description languages," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, pp. 1550–1566, Aug 2014.

[4] M. Vilaplana, E. Gallo, and F. Navarro, "Towards a formal language for the common description of aircraft intent," *24th Digital Avionics Systems Conference*, vol. 1, pp. 3.C.5–3.1–9, 2005.

[5] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.

[6] F. J. S. Nieto, "The long journey toward a higher level of automation in atm as safety critical, sociotechnical and multi-agent system," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 230, no. 9, pp. 1533–1547, 2016.

[7] Y. Chen, L. Zhou, J. Yang, and Y. Yan, "Big data platform of air traffic management," in *2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 2019.

[8] K. Legrand, D.), and C. Rabut, *Correction and Optimization of 4D Aircraft Trajectories by Sharing Wind and Temperature Information*. 2019.

[9] E. Calvo Fernández, L. Sanz, J. Cordero Garcia, and R. Valdés, "Conflict-free trajectory planning based on a data-driven conflict-resolution model," *Journal of Guidance, Control, and Dynamics*, vol. 40, pp. 615–627, 03 2017.

[10] Z. Wang, M. Liang, D. Delahaye, and W. Wu, "Learning real trajectory data to enhance conflict detection accuracy in closest point of approach," in *USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)*, pp. 1–8, 2019.

[11] J. Tang, "Conflict detection and resolution for civil aviation: A literature survey," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 10, pp. 20–35, 2019.

[12] R. Agrawal and C. Nyamful, "Challenges of big data storage and management," *Global Journal of Information Technology: Emerging Technologies*, vol. 6, no. 1, pp. 1–10, 2016.

[13] S. Ruiz, M. Piera, J. Nosedal, and A. Ranieri, "Strategic de-confliction in the presence of a large number of 4d trajectories using a causal modeling approach," *Transportation Research Part C: Emerging Technologies*, vol. 39, pp. 39:129–147, 2014.

[14] S. Ayhan, P. Costas, and H. Samet, "Prescriptive analytics system for long-range aircraft conflict detection and resolution," in *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 239 – 248, Nov 2018.

[15] G. G. N. Sandamali, R. Su, K. L. K. Sudheera, Y. Zhang, and Y. Zhang, "Two-stage scalable air traffic flow management model under uncertainty," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7328–7340, 2021.

[16] R. Sabatini, A. Gardi, K. D. Ridder, S. Ramasamy, and L. Salazar, "Automated intent negotiation and validation system for 4-dimensional trajectory based operations," in *European Navigation Conference (ENC)*, p. pp, 04 2013.

[17] K. quan Cai, Y. wu Tang, and W. Wang, "An evolutionary multi-objective approach for network-wide conflict-free flight trajectories planning," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pp. 1D2–1–1D2–10, Sep 2015.

[18] J. Rosenow, M. Lindner, and J. Scheiderer, "Advanced flight planning and the benefit of in-flight aircraft trajectory optimization," *Sustainability*, vol. 13, no. 3, 2021.

[19] S. Malakis, P. Psaros, T. Kontogiannis, and C. Malaki, "Classification of air traffic control scenarios using decision trees: Insights from a field study in terminal approach radar environment," *Cogn. Technol. Work*, vol. 22, p. 159–179, feb 2020.

[20] Y. Liu and M. Hansen, "Predicting aircraft trajectories: A deep generative convolutional recurrent neural networks approach," 2018.

[21] R. Wang, R. Alligier, C. Allignol, N. Barnier, N. Durand, and A. Gondran, "Cooperation of combinatorial solvers for en-route conflict resolution," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 36–58, 2020.

[22] J. Tang, "Conflict detection and resolution for civil aviation: A literature survey," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 10, pp. 20–35, 2019.

[23] I. R. De Oliveira, V. F. Ribeiro, L. Weigang, and R. C. Garcia, "Detecting violation of aircraft separation requirements," Feb. 4 2020. US Patent 10,553,121.

[24] J. Besada, G. Frontera, J. Crespo, E. Casado, and J. López-Leonés, "Automated aircraft trajectory prediction based on formal intent-related language processing," *IEEE Transactions on Intelligent Transportation Systems*, 2013.

[25] EUROCONTROL and FAA, "Common trajectory predictor structure and terminology," Tech. Rep. Version 1.0, Action Plan 16 White Paper, Jan 29th 2010.

[26] M. O. Ball, R. L. Hoffman, D. Knorr, J. Wetherly, and M. Wambsganss, "Assessing the benefits of collaborative decision making in air traffic management," *Progress in Astronautics and Aeronautics*, vol. 193, pp. 239–252, 2001.

[27] J. López-Leonés, M. A. Vilaplana, E. Gallo, F. A. Navarro, and C. Querejeta, "The aircraft intent description language: A key enabler for air-ground synchronization in trajectory-based operations," in *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*, pp. 1.D.4–1–1.D.4–12, Oct 2007.

[28] A. B. M. Moniruzzaman and S. Hossain, "Nosql database: New era of databases for big data analytics - classification, characteristics and comparison," *Int J Database Theor Appl*, vol. 6, 06 2013.

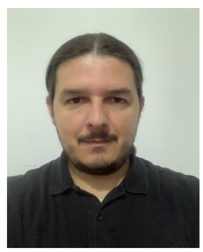
[29] Y. Li and S. Manoharan, "A performance comparison of sql and nosql databases," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pp. 15–19, IEEE, 2013.

[30] R. E. Korf, *Artificial Intelligence Search Algorithms*, p. 22. Chapman & Hall/CRC, 2 ed., 2010.

[31] IATA, "Iata forecast predicts 8.2 billion air travelers in 2037," 2018. Press Release No.: 62. Date: October 24th, 2018.

[32] A. N. de Aviação Civil, "Relatório operacional de segurança operacional (raso) - 2020," tech. rep., ANAC, 2020.

[33] DECEA, "Anuário estatístico de tráfego aéreo," tech. rep., Headquarters of Air Navigation Management (CGNA), 2018.



Lucas Borges Monteiro is a Ph.D candidate at the Computer Science Department (CiC) at University of Brasılia (UnB), Brazil and a Computer Engineer graduated from the Federal University of Goiás (UFG). He received the M.Sc. degree in informatics from the UnB. He currently works at Brazilian Internal Revenue Service (IRS). His research interests are artificial intelligence, machine learning, and air traffic management.

Vitor Filincowsky Ribeiro received the graduate degree in computer science, the M.Sc. degree and PhD in informatics from the University of Brasília (UnB), Brazil. He is a Senior Systems Analyst with the Brazilian Society for IT Development in Brasília. He graduated with a thesis on collaborative decision making. His current research interests include cooperative algorithms for resource allocation and scenario forecast with the emphasis in ATFM. He was a recipient of the Brazilian National Confederation of Transportation Prize in 2013 and also Honorable Mention Thesis of the UnB Dissertation and Thesis Award 2018 and 2019.

Cristiano P. Garcia is a M.Sc. Student at the Computer Science Department (CiC) at University of Brasılia (UnB). He received his Bachelor's degree in Computer Science from the University of São Paulo (USP) in 2009. He is also a Brazilian Air Force Captain specializing in Air Traffic Management. He currently works at CINDACTA I, a Regional Unit of the Department of Airspace Control (DECEA). His research interests are artificial intelligence, machine learning, air traffic management, and urban air mobility.

Geraldo P. Rocha Filho is an Assistant Professor at the Computer Science Department (CiC) at University of Brasılia (UnB). He received his Ph.D. in Computer Science from the University of São Paulo (USP) in 2018. He received his M.Sc. from the USP in 2014. He was also a post-doctoral research fellow at the Institute of Computing at UNICAMP before joining the UnB. His research interests are wireless sensor networks, vehicular networks, smart grids, smart home and machine learning.

Li Weigang is a full professor and coordinator of TransLab of the Department of Computer Science at the University of Brasilia (UnB), Brazil. In 1994, he received Ph.D. degree from the Aeronautics Institute of Technology (ITA), Brazil. Currently, he is a fellow researcher (PQ) from Brazilian National Council for Scientific and Technological Development (CNPq) and has coordinated various R&D projects supported by some promotion foundations and also the industry partners. He advised more than 100 students including PhD students and Postdoctoral researchers during his long career. His main research interest is Artificial Intelligence and its applications such as Intelligent Air Transportation and Machine Learning.