Aircraft Flight Plan Optimization with Dynamic Weather and Airspace Constraints

Coline Ramée, Junghyun Kim, Marie Deguignet, Cedric Justin, Simon Briceno, Dimitri Mavris
Aerospace Systems Design Lab
Georgia Institute of Technology
Atlanta, GA, USA
coline.ramee@gatech.edu

***Abstract* — Flight planning is the process of producing a flight plan which describes a proposed aircraft trajectory. This task is typically performed ahead of departure with the intent of minimizing operating costs, while accounting for weather, airspace, traffic, and comfort considerations. Recent improvements in cockpit connectivity present new opportunities for flight crews to continuously re-assess the trajectories once in the air using the latest information sets (weather observations and forecasts, traffic). In turn, this enables flight crews to proactively respond to the uncertain evolution of the weather by steering the aircraft along optimal trajectories. This also brings new challenges as flight crews are ill-equipped to continuously process vast amount of information to perform the trajectory optimization. A framework is therefore proposed to automate the fusion of various sources of information (severe weather, winds aloft, restricted airspace) to feed a trajectory optimizer that continuously updates the aircraft trajectory. This relies on the implementation of the A* algorithm with the objective to minimize cruise fuel burn and emissions. Use-cases are investigated by comparing continuously updated trajectories with actual flight trajectories retrieved from the FAA Traffic Flow Management Systems through consumer-oriented websites. Promising results are observed with fuel burn savings reaching 8%.**

Path planning; convective weather; wind; airspace

I. INTRODUCTION

Flight delays are costing the air transportation industry and society billions of dollars owing to additional operating costs, lost passenger time, airline schedule padding, forced flight rescheduling, and more generally lost productivity [1]. The Federal Aviation Administration estimates that an hour of delay costs airlines between $1,400 and $4,500 depending on the type of aircraft and whether the delay occurs on the ground or in the air [2]. There are many causes to these delays and a recent study by the Bureau of Transportation Statistics reveals that the largest source of delay is weather [3]. Indeed, roughly 38% of the total delay-minutes can be attributed to either extreme weather, weather-induced late aircraft arrivals, or weather-induced national aviation system delays. Investigating deeper the types of weather causing these delays indicates that the reason varies by season: while low ceilings and low visibility prevail in the Winter, convective weather prevails in the Summer [2]. Focusing closely on the cruise phase, leading sources of delays include convective weather, winds, icing, and turbulence. One potential avenue to mitigate flight delays is to continuously retrieve real-time data about the airspace and the evolution of the weather to proactively avoid large weather systems and congested areas.

Flight planning is the process of creating a trajectory aiming at minimizing one objective, usually operating expenditures. This is a complex multi-disciplinary process that involves aircraft performance considerations, airspace-use considerations, air-traffic congestion considerations and weather considerations. Air carriers employ dispatchers and weather forecasters to generate near-optimal routings at the time of departure [4]. As the flight progresses, the information set (winds aloft, temperature, convective weather, congestion) evolves and the predeparture trajectory may no longer be flyable, let alone optimal. Larger airliners rely on dispatchers on the ground to re-assess trajectories, but the process may be slow and tedious. Smaller operators, corporate aviation, and general aviation typically do not have dispatchers, and flight crews need to reassess the trajectory when conditions change significantly. Generating optimum trajectories ‘on the fly’ while under high workload conditions in the cockpit is a daunting task. A need has thus been identified to automate the continuous re-planning of flight trajectories. This is enabled by the current convergence of technologies in terms of weather product digitization, improved on-board computational power, and improved cockpit connectivity through broadband communication. The continuous re-planning objective is to optimize trajectories to minimize operating costs and carbon emissions while concurrently reducing the flight-crew workload.

This paper describes the implementation of a continuous re-planning tool named RTOP (Real-time Trajectory OPtimization). The second section of this paper reviews prior art. The third section describes how airspace and weather data from various sources are fused together to provide real time data pertinent for the flight planning exercise. The fourth section details the modeling required to generate trajectories using up-to-date information sets. The fifth section details the optimization of trajectories. Finally, the sixth section highlights various use-cases to benchmark the proposed approach.

II. EXISTING WORK AND PROPOSED APPROACH

Studies have shown that there is a clear benefit for airlines to using wind-optimal approach to reduce fuel consumption and travel time compared to flying fixed routes [5], [6]. This is a problem difficult for humans to solve and there have been multiple types of approaches proposed to automatically find solutions.

The first type of approaches formulates the problem as a variation of Zermelo's navigation problem, an optimal control problem [7]. The primary goal of these approaches is to find the best path through a wind field. Other constraints can be introduced by the mean of penalty functions as explained in [8]. The problem can then be solved using Pontryagin's Minimum Principle. The solutions obtained by these approaches are the aircraft's optimal heading throughout the flight. However, although continuous approaches would work well in a free-flight paradigm, they are not realistic in the current traffic management context where flight plans are defined as a list of waypoints that the aircraft is expected to fly straight to or predefined routes that the aircraft must follow.

Local approaches focus on updating a pre-existing flight plan based on new information such as convective weather or traffic. The Traffic Aware Planner (TAP) was developed as a cockpit resource that fuses information from weather, aircraft performance, waypoints information and nearby traffic to offer pilots alternative routes that would avoid conflicts and bad weather or reduce fuel consumption or travel time. The optimizer chooses between a list of local maneuvers using a genetic algorithm [9].

Another class of approaches relies on a discretization of the problem to represent possible trajectories as a graph and uses the A* algorithm to find the shortest path through the network. Additional details on the A* algorithm can be found in section V. The framework presented here relies on such approaches. Distinct strategies within this class of approaches mostly vary by how the graph or network is built and the constraints incorporated. In [10], the environment is discretized as a grid and higher cost is associated to nodes with bad weather. In the PARTNER tool introduced in [11], the network is generated by discretizing the aircraft commands. These two approaches suffer from the same issue as the continuous approach: the paths might not be approved by ATC. The company Mosaic software creates a Clearable Route Network (CRN) using historical flights information, that they then use to generate operationally acceptable flight plans [12]. Schilke and Hecker propose in [13] the idea for a system architecture that would be weather-aware and use A* to find the shortest path. However, there is no actual implementation and they do not detail how wind and dynamic aspects would be integrated with A*. In [14], the altitude and horizontal paths are decoupled. A* is used to optimize 2D paths on a grid or using AIRAC waypoints and the resulting path is fed to another optimization module. Dynamic aspects are ignored in their implementation of A*, and the weather is assumed static during the 2D flight path optimization step.

In this paper, A* is used to find the optimal path in a 4D network that considers the dynamic aspects of the weather, constraints due to waypoints and routes and aircraft performance. We fuse data and compare to actual flight to estimate the benefits of such a system on long domestic flights over the continental US.

III. DATA COLLECTION

In order to create a real-time path planning framework, a large data amount about the flight environment must be collected: information about the weather, the aircraft, and the airspace.

A. Aircraft Performance

To determine the best operating speed and cruise altitude for an aircraft at a given weight, an aircraft performance model is required. The aircraft operating manual contains tables representing the fuel flow, operating speed and rate-of-climb (ROC) in cruise, climb and descent at different altitudes. With this information operators can select the best-range cruise altitude. However, in general these manuals are not available. In this study, a tool called FLOPS (Flight Optimization System) was used to recreate these tables. FLOPS is an aircraft synthesis software developed by NASA which can be used to simulate a specific mission profile [15].

In order to obtain a FLOPS-independent framework, a surrogate model of the performances of the aircraft is created. FLOPS is first run to generate aircraft performance datapoints in cruise, climb, and descent for a fixed Mach number. FLOPS inputs include detailed information about all the aerodynamic characteristics of the aircraft, the engine, the weights, and the

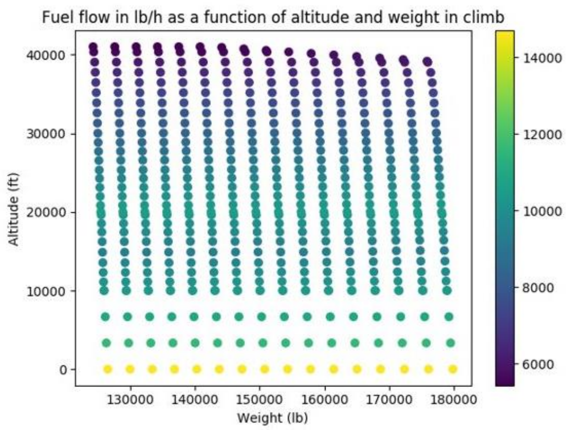

Figure 1: Illustration of the point-based space exploration for the climb portion of the flight using FLOPS

flight sequence.

Then, to explore a large continuous altitude-weight space, 2D-interpolation is used to model aircraft performance in terms of fuel flow, airspeed, and rate-of-climb (ROC) as a function of the aircraft's total weight and altitude. This is done for an aircraft similar to an A320neo. After the interpolation, a continuous performance model is generated for a fixed aircraft and a fixed Mach number.

B. Wind

The most well-known weather model products such as Global Forecast System (GFS) [16] and Rapid Refresh (RAP) [17] were considered. The High-Resolution Rapid Refresh (HRRR) model was finally chosen because:

1) the HRRR model is commercially open,

2) the HRRR model provides weather information of the Contiguous United States (CONUS) territory, and

3) the HRRR model has the highest resolution among weather model products.

The HRRR model is a National Oceanic and Atmospheric Administration (NOAA) real-time 3km resolution and hourly updated weather model produced by the National Centers for Environmental Prediction (NCEP) [18]. The HRRR model provides a set of detailed weather-related properties such as temperature and wind speed against longitude, latitude, altitude, and time. Among various HRRR weather datasets, the authors only focused on the eastward and northward wind (measured in m/s) and went through data preprocessing steps by using the Python library Pygrib [19]. Figure 2 shows an example visualization of the HRRR northward wind at a specific time and altitude.

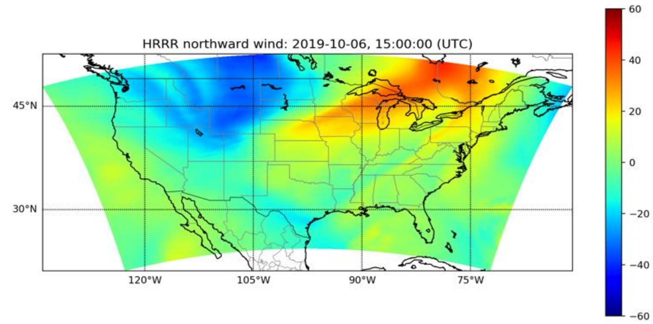

Figure 2: HRRR northward wind visualization at 2019-10-06 15:00 UTC (altitude = 250 hPa)

The information contained in the HRRR model is discrete. The wind values are available once per hour and are divided by pressure altitude. For a specific time at a specific pressure altitude the wind values are stored as a 2D table, and the corresponding latitude and longitude are stored in two tables of the same size. In a flight path planning framework, it is necessary to create a wind model that provides continuous information with respect to four-dimensional flight trajectory (i.e., timestamp, altitude, latitude, and longitude). A Quadrilinear interpolation is performed to obtain wind values anywhere in the four-dimensional space.

C. SIGMET

The framework presented here focuses on convective weather (thunderstorms), which are responsible for a majority of weather-related delays in the United States in the summer. Thunderstorms usually extend very high in altitude and impact commercial flights.

The Aviation Weather Center (AWC), which is a part of NOAA National Weather Service (NWS), publicly issues weather alerts in the form of either an Airmen's Meteorological Information (AIRMET), non-convective Significant Meteorological Information (SIGMET), or convective SIGMET for the contiguous 48 states. The convective SIGMET product is human-drawn polygons that denote regions of current convective weather that may be potentially hazardous to aircraft. The convective SIGMET is issued hourly on a scheduled basis when the following conditions are expected to occur:

1) a line of thunderstorms at least 60 miles long with thunderstorms affecting at least 40% of its length,

2) an area of active thunderstorms judged to have a significant impact on the safety of aircraft operations covering at least 40% of the area concerned and exhibiting a very strong radar reflectivity intensity,

3) embedded or severe thunderstorms expected to occur for

more than 30 minutes during the valid period regardless of the size of the area, and

4) a special case may be issued when wind gusts greater than or equal to 50 knots are reported [20].

Figure shows an example visualization of the convective SIGMET at the specific date. Convective SIGMETs are represented as convex polygons and they usually include information in terms of an initial position, a velocity, and a validity period. In general, they are valid for two hours after they are published.

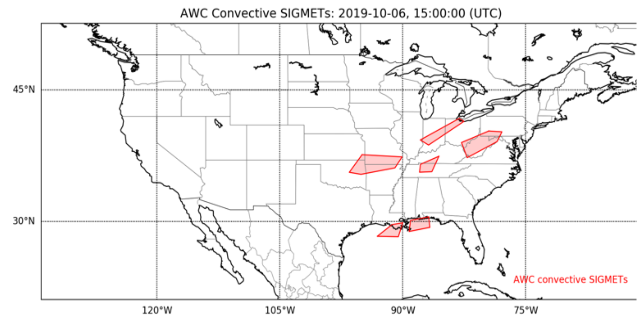

Figure 3: Convective SIGMET visualization at 2019-10-06 15:00 UTC

Although AWC publicly issues the convective SIGMET products, they have limited access to historical data. For this reason, a Python code was developed which automatically connects to the AWC TDS at regular intervals to build a database for convective SIGMET polygons.

D. FAA Waypoints and Routes

The FAA's Aeronautical Data Delivery Service is used to retrieve FAA waypoints and routes and to create the network used to define paths in flight plans [21].

Airlines and pilots are not restricted to routes and waypoints when planning a flight. An analysis of several transcontinental flights was conducted and showed that, when there is no significant weather, flights stay on a route for on average 80% of the total flight. When there is significant weather (SIGMET), that number is reduced to an average of approximately 50% of the total flight. The results presented in section VI show that even though the trajectories generated by the framework are more constrained than the reality they still improve the fuel consumption and trajectory length in most cases.

The choice of using waypoints and routes was driven by the fact that when communicating with ATC over radio, using named waypoints rather than coordinates would be easier for the pilot. It would also allow the route to be compliant with ATC requirements and therefore to be easily certified. Alternatively, the framework was setup to use a grid.

A waypoint is a predetermined geographical position defined in terms of longitude and latitude coordinates that can be used for route definition. Waypoints are often named and can be used by ATC and pilots to specify a modification in direction, speed or altitude along the current path. In the database published by the FAA, each waypoint has many different attributes such as latitude, longitude, type (RPT: Reporting Point, WPT: Waypoint, RNAV: RNAV Waypoint, … [22]), and a unique identifier in the dataset. Figure 4 depicts the density of the waypoint distribution over the continental US.

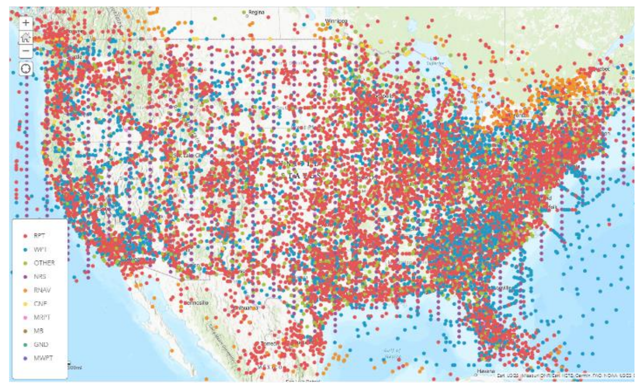

Figure 4: FAA waypoints over the United States [21]

A route or airway has no physical existence; it is a corridor that connects two specific locations at a specific altitude. In order to fly on a route, the aircraft must meet all the requirements of the airway. In the FAA database, each route has many different attributes such as a start and end waypoints (referenced by their unique identifiers as mentioned above), an altitude (high, low or both), a type (CONV: Navaid Based Route, RNAV: Area Navigation, … [22]), and its unique identifier in the dataset. Figure 5 shows the network of routes over the United States.

In the framework presented here the waypoint and route databases have been locally downloaded, and the routes have been cross-referenced with the waypoints to find the latitude and longitude at which they start, and identify which routes are connected by a common waypoint. This creates a graph network over the United States where routes are edges and waypoints are nodes. The data can then be filtered based on the attributes of the waypoints and the routes depending on the aircraft, the flight, and its characteristics to ensure that the right data is used to create the flight plan. For example, when dealing with commercial aircrafts, high altitude routes must be selected

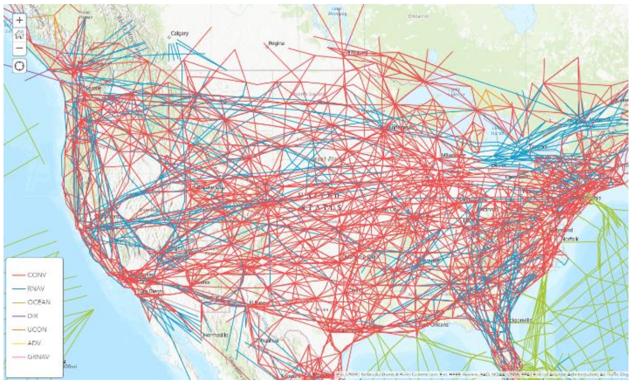

Figure 3: FAA routes over the United States [21]

to comply with ATC requirements.

The initial and final point of a trajectory are added to the list of waypoints. Since there are no route going to or from these points, nearby waypoints are queried and considered reachable. To find neighboring waypoints quickly, all waypoints are stored in a ball tree created using the scikit-learn Python library [23].

IV. MODELING

The framework integrates airspace, weather, and vehicle performance to create a united data structure that can be used to model trajectory length, fuel cost, and feasibility.

A. Assumptions

The framework relies on the following assumptions:

- Aircraft are assumed to cruise at a constant Mach number
- Traffic, turbulences, approach and departure procedures are ignored
- Convective weather is represented by SIGMETs which cannot be penetrated by the aircraft
- Aircraft's paths are constrained to be on the network of waypoints and routes defined by the FAA except at the beginning and end of the path

B. Travel Cost Function

To associate a cost to traveling along one edge of the graph, a travel cost function is required. Given departure time and aircraft weight, the travel cost function computes how long it would take and how much fuel is needed to travel between two connected nodes.

Because the two nodes can be quite distant, the curvature of the Earth must be accounted for. The trajectory between the two nodes is discretized every 40 nautical miles. To compute the bearing and positions along the great-circle distance paths, the formulas from [24] were used. The bearing of the aircraft is assumed to be constant on the discretized segment and an equirectangular projection is used to convert between latitude/longitude and a local cartesian reference frame. The aircraft true airspeed (TAS) and fuel flow are determined using the performance model based on the aircraft weight and altitude. The wind vector is estimated at the start of the segment using the wind model at the given time and altitude. The aircraft heading required to counter the wind and stay on the ground track is determined using [25] and the resulting ground speed is computed. With the ground speed and the length of the segment known, the time to complete the segment can be found. The total fuel used on the segment is obtained by multiplying fuel flow with time. The weight of the aircraft is updated, and the same operations are repeated for the next segments. This yields the total time and fuel required for the aircraft to travel between the two nodes.

To account for climbs and descents, the altitude of the two nodes is compared. If both nodes are at different altitudes, an additional discretization step is performed at the beginning. The altitude is discretized in 1,000ft increments. The rate of climb, fuel flow, and true airspeed are computed as functions of altitude and aircraft weight and a procedure similar to the one explained previously is conducted until the aircraft reaches the desired altitude, then the remaining cost to the destination node is computed as before.

C. Availability Function

In addition to the travel cost function, an availability function is needed. Given departure time and aircraft weight, the availability function evaluates whether traveling between two connected nodes will result in a collision with an obstacle. The positions of the aircraft and of the obstacles can be represented by piecewise linear functions, since their velocities are constant along time segments. To find the time of closest approach of two objects A and B with constant velocity the formula can be easily derived:

$$t_{CPA} = -\frac{(\overrightarrow{V_B} - \overrightarrow{V_A}).(\overrightarrow{P_B} - \overrightarrow{P_A})}{\left\|\overrightarrow{V_B} - \overrightarrow{V_A}\right\|^2}$$

Where $\overrightarrow{V_A}$ and $\overrightarrow{V_B}$ are the velocities of the objects, $\overrightarrow{P_A}$ and $\overrightarrow{P_B}$ are the positions of the objects at time t_0 projected on a local cartesian frame, and t_{CPA} is the time from t_0 when the objects are closest to each other.

Since the analysis is conducted in a piecewise manner, the time is clamped on the time interval where both objects exist and have constant velocities. A first check is done using the centroid and maximum radius of the polygon. If at the time of closest approach, the distance between the two objects is greater than the radius of the polygon, there is no collision on

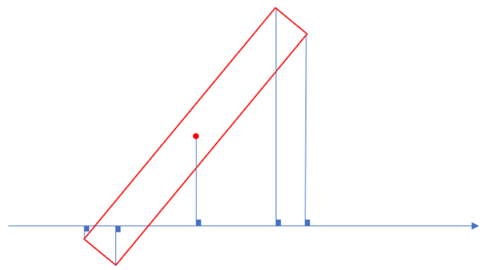

Figure 5: Illustration of the points of closest approach between a polygon with zero velocity and an object moving along the direction of the arrow

that time segment. If the distance is smaller, then there is no simple way to check. A point in polygon check is performed using the Shapely Python library [26]. However, it is not enough to perform the point in polygon check at the time of closest approach of the centroid to know if there is a collision. As illustrated on Figure 6, the time of closest approach must be determined for all vertices of the polygon and the point of closest approach checked for inclusion. The check is performed for all objects that have been forecasted and that exist in the time segment from when the aircraft leaves the first node to when it reaches its destination node. There are some approximations due to the conversion from spherical to cartesian geometry.

In the model proposed here, SIGMETs represent hard constraints that the aircraft should always avoid. In reality, SIGMETs can sometimes be penetrated as their boundaries are defined conservatively and there can sometimes be a path clear of storms that goes through them. While restricted airspace could also be included in the list of obstacles, they were neglected since they usually extend to altitudes significantly lower than usual commercial cruise altitudes.

D. *Multilinear Interpolation*

Both the data from the aircraft performance model and the wind are provided as discrete datasets and structured roughly as a grid. However, the travel cost and availability functions must query aircraft performance and wind values at any point within the envelop. The Scipy 2D interpolation function was used to build functions that can be queried at any point [27]. In order to query the wind at any altitude and time, the 2D interpolation function was wrapped to create a 4D interpolation function.

E. *Validation*

The travel cost function for time was validated by comparing actual flights duration to the duration obtained by running these flight paths in the proposed model. The actual flight trajectories (altitude, latitude, longitude and time) were retrieved from the Flight Aware website. Validation cases were selected such that

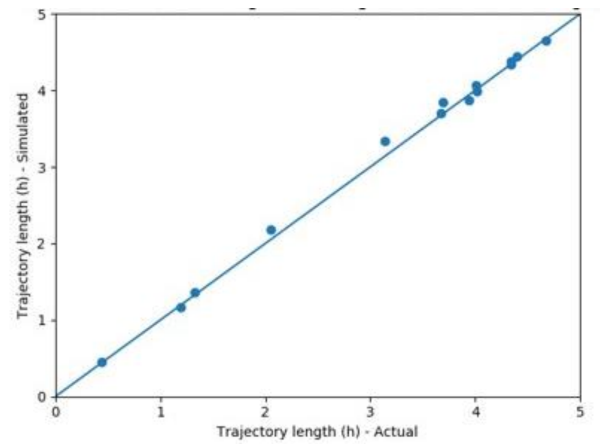

Figure 4: Simulated versus actual flight time using the aircraft model

they covered days marked by weather events (e.g. thunderstorms) and long flights (three hours or longer), but also some shorter flights to account for the natural variety of flights. The 14 selected validation flights were operated by Spirit Airlines on A320neo aircraft. over the continental US on 9 different days. Corresponding historical weather data was retrieved for those days. The analysis focused on the cruise portion of the flight, since departure and approach are much more constrained. The trajectories were cropped to only keep the portion of the flights above 31,000ft. The results, illustrated on Figure 7, show a good agreement between simulation and reality for flight time, which validates the aircraft speed model and the wind model. The fuel model could not be validated due to the lack of publicly available data.

Because the initial weight of the aircraft is not publicly available, the take-off weight was computed by making a few assumptions for the purpose of the analysis. The aircraft weight when landing was determined by assuming a full flight with cargo and passengers and the reserve fuel required by regulations. The fuel required for the flight is estimated using the great-circle distance and an estimate of fuel burn at a reasonable cruise altitude of 33,000 ft. The weight at take-off is estimated to be the sum of the arrival weight and trip fuel. This initial estimate does not consider wind. This could be addressed by running the algorithm a first time to get an estimate of the fuel required for the trip and iterating the analysis.

V. OPTIMIZATION

The final structure of the model is a graph whose nodes are associated to a waypoint and an altitude. Pairs of nodes are connected if and only if their respective waypoints are connected by a route. The availability function is used to check if a SIGMET could prevent the aircraft to fly on that route. The travel cost function estimates the fuel and time taken by the aircraft to travel between the two nodes.

The A* algorithm is used to perform flight planning in this four-dimensional space, its goal is to minimize fuel cost.

The A* algorithm can be shown to be [28]:

- Complete: the algorithm will find a solution if there is one
- Optimal: the algorithm will find the shortest path if the heuristic used is consistent
- Optimally efficient: given the same heuristic no other algorithm can be guaranteed to expand fewer nodes

These three qualities explain why A* is such a popular algorithm to optimize discrete trajectories. A* performance degrades as the number of neighbors of a state increase. Here, optimizing a cross-country flight takes on the order of ten minutes on a laptop PC.

The heuristic, i.e. an estimate of the minimum amount of fuel required to reach the destination, is computed in several steps. First the remaining distance to the goal is estimated using the great circle distance (haversine formula). Then, an optimistic estimate of the aircraft velocity is required. Since the goal is to optimize for fuel and not for time, the velocity and fuel flow selected are those that maximize the aircraft distance per pound of fuel. These values are found using the FLOPS data. To account for wind, the maximum current tailwind is estimated. The heuristic choice greatly impacts the performance of the algorithm in terms of number of expanded nodes and hence runtime.

VI. RESULTS

The algorithm was run on the 14 test cases. Figure 8 shows the result of the optimization when run on an example flight operated by Spirit Airlines from Cleveland to Los Angeles. For that case, the original and optimized trajectories are similar. The optimized altitude is lower than the actual altitude at which the aircraft flew. This might be due to a difference between the initial weight estimate and the actual weight of the aircraft. The initial weight was estimated to be 168,562 lb. for that flight using the method explained in section IV.E. According to the A320 Flight Crew Operating Manual, the best altitude for the aircraft around that weight should be 33,000ft. However, since the aircraft was heading west, it was constrained to operate at an even flight level such as 34,000ft or 36,000 ft. Using the travel cost function on its trajectory the actual flight is estimated to have burnt 22,179lb of fuel, whereas the optimized trajectory would have burnt an estimated 21,853lb of fuel. This corresponds to a 1.5% reduction in fuel burn. The actual cruise portion of the Spirit flight took 4.34 hours. Keeping its trajectory the same and modeling the flight with our aircraft model (similarly to what was done in IV.E for validation), a flight time of 4.38 hours is found. When analyzing the trajectory proposed by A*, the flight time is 4.34 hours. The time length change is computed on the path obtained with the aircraft model to be consistent with the fuel computation, and results in a 0.9% reduction in travel time, which is not significant.

Figure 9 shows a scatter plot of the performance of the optimized trajectory relative to the original trajectory for each test case. Each axis shows the percentage of change of the new trajectory compared to the original trajectory. The routes chosen by the algorithm are shown to improve fuel burn by a few percent and improve the duration of the flight in most cases. Over the 14 cases the average fuel reduction is -3.4% and the 95% confidence interval is $[-5.6; -1.1]$. For time, the

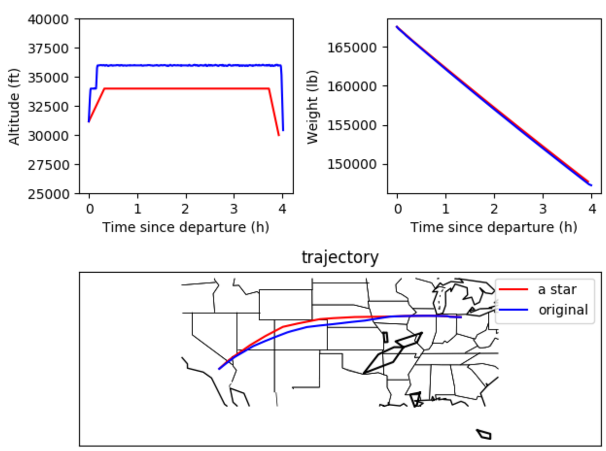

Figure 8: Results for flight NKS185 on October 10, 2019

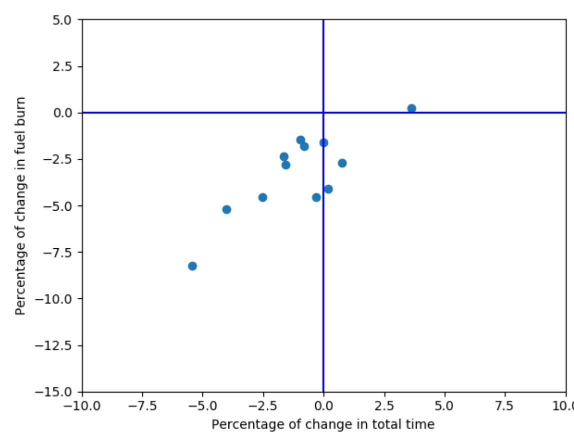

Figure 9: Comparison of the optimized trajectory against the actual trajectory in terms of fuel burn and time

average improvement is only -0.8 and the 95% confidence interval $[-3.9; 2.3]$.

Cases where the algorithm fails to improve or match the performance of the original trajectory may be due to several factors. First, if the cruise section of the flight is very short (around 1h), constraining the aircraft to fly on routes severely limit possibilities and leads to sub-optimal flight plans. Second, the SIGMETs forecast used in the model are limited to a 2h window, whereas airlines have access to more detailed weather forecast. For example, in one case, a thunderstorm line extended very widely from north to south and caused the algorithm to pick an optimistic route, working under the assumption that thunderstorms would dissipate. Because the thunderstorm remained, the aircraft finally had to make a detour. On the other hand, the airline picked a trajectory that avoided the thunderstorm line from the start of the flight onwards because it had access to a better weather forecast.

VII. CONCLUSION

The preliminary RTOP framework integrates weather, airspace rules and aircraft performance and can be used to model and optimize flights. The optimization algorithm runs fast enough that it could be run multiple time during the flight to account for updated weather information, and help pilots request updated paths from Air Traffic Control. Large airlines can rely on dispatchers, flight specialists, and meteorologists to optimize flights. An automated tool such as the one proposed here would allow smaller companies or business jets to perform the same optimization for a fraction of the cost.

There are many elements that could be improved to increase the accuracy of the solution. Integrating traffic or turbulence information in a manner similar to the SIGMETs would not change the complexity of the algorithm and would more accurately reflect commercial aircraft constraints.

REFERENCES

[1] M. Ball et al., "Total Delay Impact Study: A comprehensive assessment of the costs and impaocts of flight delay in the United States," 2010.

[2] "NextGen Weather, FAQ: Weather Delays," *2013 Operations Network Report, Federal Aviation Administration*, 2013. [Online]. Available: www.faa.gov/nextgen/programs/weather/faq/.

[3] "Understanding the Reporting of Causes of Flight Delays and Cancellations," *Bureau of Transportation Statistics*, 2019. [Online]. Available: www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations.

[4] M. C. Dorneich et al, "Integration of Weather Information into the Dispatcher Pre-Flight Route Selection Process," in *Proc. of the 21th Digital Avionics Systems Conference*, 2002.

[5] K. Palopo, R. D. Windhorst, S. Suharwardy, and H. T. Lee, "Wind-optimal routing in the National Airspace System," *J. Aircr.*, vol. 47, no. 5, pp. 1584–1592, 2010.

[6] B. Sridhar, H. K. Ng, F. Linke, and N. Y. Chen, "Benefits analysis of wind-optimal operations for trans-atlantic flights," in *AIAA AVIATION 2014 -14th AIAA Aviation Technology, Integration, and Operations Conference*, 2014, no. June, pp. 1–12.

[7] M. R. Jardin and A. E. Bryson, "Methods for computing minimum-time paths in strong winds," *J. Guid. Control. Dyn.*, vol. 35, no. 1, pp. 165–171, 2012.

[8] H. K. Ng, B. Sridhar, S. Grabbe, and N. Y. Chen, "Cross-polar aircraft trajectory optimization and the potential climate impact," in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, pp. 1–19.

[9] S. E. Woods, R. A. Vivona, D. A. Roscoe, B. C. LeFebvre, D. J. Wing, and M. G. Ballin, "A cockpit-based application for traffic aware trajectory optimization," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, pp. 1–14.

[10] D. Knapp, T. Jameson, E. Measure, and A. Butler, "Optimized Flight Routing Based on Weather Impacts Grids," in *13th Conference on Aviation, Range and Aerospace Meteorology*, 2008, vol. 316, no. 10, p. 1116.

[11] J. Kirk, S. Balasekar, S. Tarazi, M. Hassan, M. Kirby, and D. Mavris, "Parametric real-time navigation en-route," in *AIAA Scitech 2019 Forum*, 2019, no. January, pp. 1–21.

[12] "Aircraft Trajectory Optimizer." [Online]. Available: https://www.mosaicdatascience.com/2017/09/19/predictive-iot-services-aircraft-trajectory-optimizer/.

[13] C. Schilke and P. Hecker, "Dynamic route optimization based on adverse weather data," in *SIDs 2014 - Proceedings of the SESAR Innovation Days*, 2014, no. November.

[14] M. Lindner, J. Rosenow, and H. Fricke, "Dynamically Optimized Aircraft Trajectories Affecting the Air Traffic Management," in *International Conference on Research in Air Transportation*, 2018, no. July.

[15] D. P. Wells, B. L. Horvath, and L. A. McCullers, "The Flight Optimization System Weights Estimation Method," Hampton, VA, 2017.

[16] "Global Forecast System." [Online]. Available: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs.

[17] "Rapid Refresh (RAP)." [Online]. Available: https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/rapid-refresh-rap.

[18] "The High Resolution Rapid Refresh (HRRR)." [Online]. Available: https://rapidrefresh.noaa.gov/hrrr/.

[19] J. Whitaker, "Pygrib." 2010.

[20] M. Graf and B. Entwistle, "En Route Forecast and Advisories," 2019.

[21] "FAA's Aeronautical Data Delivery Service." [Online]. Available: https://adds-faa.opendata.arcgis.com/.

[22] FAA Aeronautical Information Services, "AIS Open Data: Data Structure and Organization," 2019.

[23] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 1, pp. 29–33, 2011.

[24] C. Veness, "Calculate distance, bearing and more between Latitude/Longitude points." [Online]. Available: https://www.movable-type.co.uk/scripts/latlong.html.

[25] R. Nave, "Aircraft Heading to Conter Wind," *Hyperphysics*. [Online]. Available: http://hyperphysics.phy-astr.gsu.edu/hbase/lcos.html#c3. [Accessed: 14-Jan-2020].

[26] S. Gillies and others and toblerity.org, "Shapely: manipulation and analysis of geometric objects." 2007.

[27] P. Virtanen *et al.*, "SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python," *arXiv Prepr.*, pp. 1–22, 2019.

[28] S. J. Russel and P. Norvig, "A* search: Minimizing the total estimated solution cost," in *Artificial Intelligence: a Modern Approach*, Third Edit., Prentice Hall, 2009, pp. 93–99.