A Waypoint-Based Aircraft-Rerouting Routine for Severe Weather Events

Colin Shi*, Preston Wang†, and Nathaniel J. Wei‡
Stanford University, Stanford, CA, 94305, USA

The rerouting of aircraft approaching an airport around a local storm is critical for minimizing flight delays and revenue losses while prioritizing flight safety. This multi-objective planning problem is made tractable by discretizing the space around the airport into a grid of waypoints for the aircraft. The waypoint-based method for aircraft rerouting leaves the aircraft and storm unconstrained by the grid in their movements, while significantly reducing the sizes of the state and action spaces that the flight controller must consider. This problem is solved for a single aircraft and a model storm using two different control schemes: a Markov decision process (MDP) policy, and the A* weighted shortest-path algorithm. These models are compared in simulations to evaluate the performance of offline and online solution methods in this novel problem formulation for the aircraft-rerouting problem. Though the MDP-based solver results in fewer and less significant flight delays than the A* algorithm for all experiments presented in this work, the A* solver may generalize more favorably to larger state and action spaces that would be prohibitively expensive for the MDP solver to compute.

I. Introduction

Severe-weather events are a significant cause of flight delays in commercial aviation [1], and the rerouting of aircraft in response to storms is therefore a critical optimization problem for air-traffic control in the vicinity of major airports. While traditionally, aircraft rerouting has been handled manually in real time, state-of-the-art decision-making algorithms have shown promise in automating these tasks, offering potential improvements in both efficacy and runtime performance. The Markov decision process (MDP) framework is an appealing control model for achieving optimal solutions to this routing problem, particularly because it is possible to generalize to the case of state uncertainty through a partially observable Markov decision process (POMDP). These frameworks have recently been applied effectively to aircraft-rerouting problems in several contexts (e.g. [2–4]).

A major obstacle to the implementation of MDP-based frameworks is the size of the state and action spaces involved. In the problem of aircraft rerouting, the continuous space of aircraft and weather dynamics can be discretized in several ways. For example, the aircraft and weather event can be constrained to move on a square grid. This formulation, used by [2], facilitates the selection of optimal paths to a specified location, but does not generalize well to real aircraft flight patterns. An alternative discretization is to limit the action space of the aircraft to discrete heading adjustments, as employed by [3] and with a modified discretization scheme by [4]. This formulation is more realistic, but the method for path optimization is not as straightforward as is the case for a grid-based formulation.

The current work thus combines grid-based and heading-based aircraft-control schemes, in order to optimally route aircraft around weather events to a target airport. The combined scheme seeks to produce more realistic flight trajectories and to more readily generalize to real aircraft-control systems. The weather event is modeled as a single storm with a given cost distribution and propagation model, which can easily be generalized to arbitrary storm types and characteristics. The problem is formulated as a multi-objective optimization problem, in which the cost of flying through the storm must be balanced with the cost of deviating from the shortest path to the airport. This novel problem formulation is solved using an MDP solver and a weighted shortest-paths algorithm, in order to evaluate the efficacy and flexibility of the formulation and to compare the performance of offline and online solution methods. The results of simulations with these two control schemes are presented, and the characteristics and relative merits of the solution methods are evaluated.

*M.S. student, Mechanical Engineering, Stanford University
†M.S. student, Aeronautics and Astronautics, Stanford University
‡Ph.D. candidate, Mechanical Engineering, Stanford University

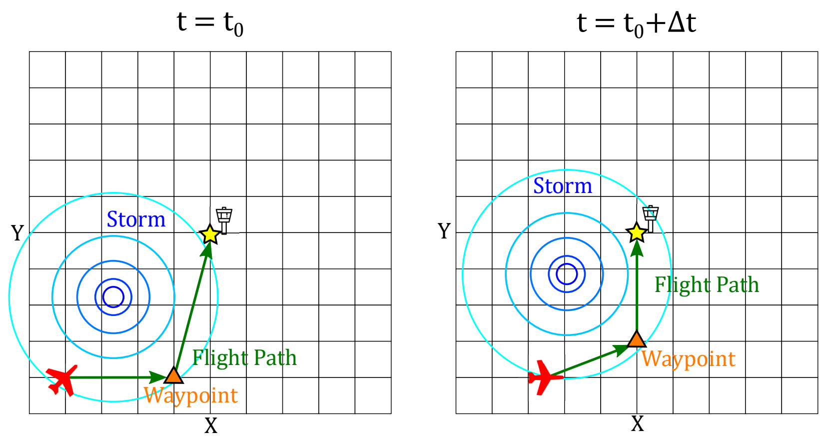


Fig. 1 Schematic of the problem setup for an arbitrary flight path at a given time-step. The grid represents the possible locations of the aircraft, storm center, and waypoint in the MDP formulation of the problem. During execution, the aircraft and storm are not constrained to the discrete grid points.

II. Problem Formulation

The problem is formulated on a square grid of $M \times M$ discrete points, where $M = 11$ for this study. The length of each side is 100 miles, corresponding to a grid spacing of 10 miles. Given that the goal of this work is to investigate a more realistic approach to aircraft rerouting around a storm, the plane is not constrained to move along a particular grid with discrete headings. Instead, the plane can move anywhere within the two-dimensional environment. The plane is modeled as a 2D kinematic object with position (x_p, y_p), heading θ, and constant velocity v of 3 miles per minute (corresponding roughly to the approach speed of a Boeing 737 aircraft).

The storm is modeled as a 2D Gaussian distribution,

$$S(x, y) = \exp\left(-\frac{(x - x_s)^2 + (y - y_s)^2}{2\sigma^2}\right), \tag{1}$$

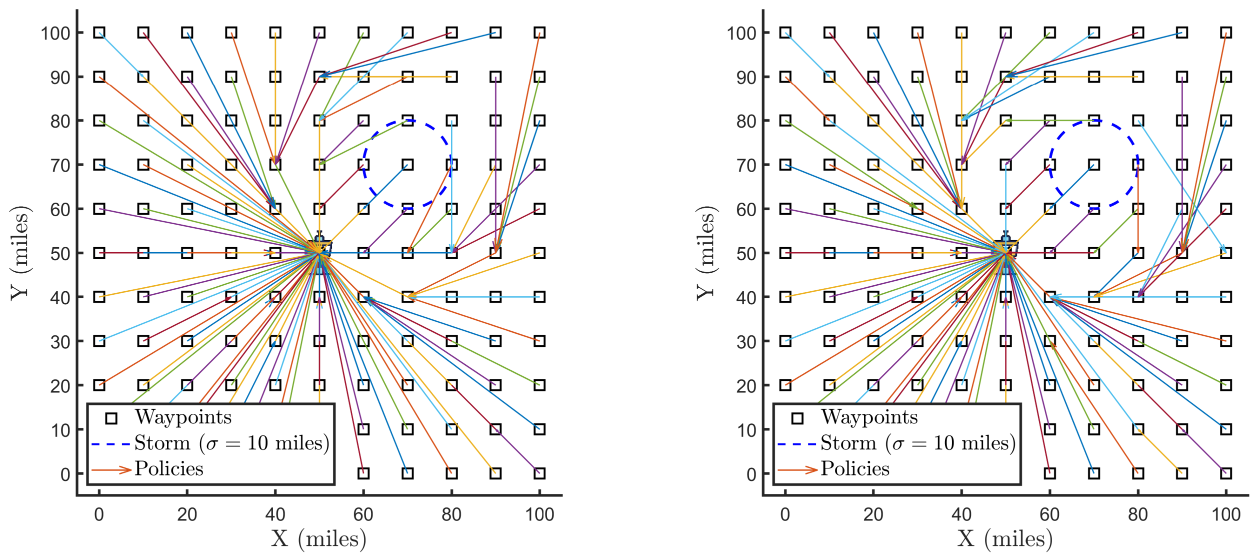
defined by its location (x_s, y_s) and standard deviation σ. The value of the distribution at a given location (x, y) represents the penalty that an aircraft would incur if it were at that location in the storm. The storm moves at a constant velocity u in directions determined by a specified distribution of transition probabilities. For example, the storm can move in random directions according to a uniform distribution, follow a single specified direction with probability 1, or behave according to distributions gleaned from historical data. The generality of this approach allows for more sophisticated storm models to be implemented in simulations [e.g. 5]. The storm velocity is set at $u = 0.75$ mi/min, in accordance with typical convective wind velocities of multicellular thunderstorms [6].

To reroute the airplane, the controller issues target waypoints for the plane to head towards. These waypoints, located at points in the $M \times M$ grid, are updated by the controller based on the current positions of the plane and storm, in order to route the aircraft around the storm. Thus, for any given aircraft location, there are $M^2 = 121$ possible waypoints available for selection. For the purposes of tractability, one waypoint is specified at each time-step, chosen by the controller from the set of M^2 waypoints, and the aircraft is assumed to proceed from this waypoint directly to the airport. A schematic example of this process of waypoint selection is given in Fig. 1. This methodology of waypoint-based navigation aligns with proposed unmanned traffic management systems involving unmanned aircraft systems (UAS) [7, 8].

The problem is formulated as a multi-objective optimization problem. The objectives are as follows: (1) to minimize the length of the flight path, (2) to minimize the cost incurred by flying through the storm, and (3) to reach the airport. The cost function is computed from the weighted sum of the total distance traveled by the aircraft along the planned flight path $\mathcal{P}$ and the line integral of the flight path through the Gaussian distribution of the storm. This can be written as

$$C(\mathcal{P}) = -\left(\alpha \int_{\mathcal{P}} ds + \beta \int_{\mathcal{P}} S(x, y) ds\right), \tag{2}$$

where α and β are the weights for the distance and storm costs, respectively. The distance weight is set to 1. The equivalent storm weight is then derived by calculating the storm cost $\int_{\mathcal{P}_\sigma} S(x, y) ds$ associated with taking a circular



(a) (b)

Fig. 2 Policies for waypoint selection, represented by arrows at each grid location, for a storm located at $(X_s, Y_s) = (70, 70)$ miles with (a) standard storm weights and (b) doubled storm weights in the cost function.

path $\mathcal{P}_\sigma$ with radius σ around the storm, defined on $(x - x_s)^2 + (y - y_s)^2 = \sigma^2$. Equating this cost to the length of $\mathcal{P}_\sigma$ yields a correction factor to the storm weight of $e^{1/2}$, so that $\frac{\beta}{\alpha} = e^{1/2}$ represents equivalent storm and distance costs. The storm weight is then multiplied by 2 to increase the cost of flying through the storm relative to the corresponding distance cost. Lastly, a large reward for reaching the airport is provided for the MDP algorithm to ensure that the optimal policy always routed the aircraft toward the airport.

III. Algorithm Design

A. MDP Algorithm

The problem formulation can be characterized as a Markov Decision Process where the action is the waypoint location, the states are the plane and storm positions, the associated costs are defined in Eqn. 2, and the reward is defined by the plane reaching the airport. In this case, the aircraft transitions are completely deterministic. Given that approach air traffic controllers route a large number of aircraft in a congested airspace, the algorithm should quickly determine the optimal policy for each aircraft. Solving the problem offline and using a policy look-up table would allow the controller to quickly determine the optimal policy. However, this presents an intractable problem because both the plane and the storm's positions are continuous functions. Therefore, the plane and storm positions are discretized to coincide with the waypoint locations, resulting in a state space of size M^4 for all possible plane and storm combinations. The policy for plane and storm positions between grid points is determined at runtime through rectangular interpolation to the closest grid points. In the case where the plane or the storm leaves the grid, the algorithm uses the policy for the closest point on the grid. The horizon is also limited to a depth of 1, corresponding to the plane heading directly towards the airport after reaching the target waypoint. This significantly reduces computational complexity, and also implicitly minimizes the number of waypoints given to the airplane. To solve the MDP problem, a MATLAB-based value-iteration solver written by [9] is employed, with the discount factor set as $\gamma = 0.95$ and the termination requirement established by $\epsilon = 0.01$. Based on the set of provided storm and cost parameters σ, α, and β, the MDP solver calculates the associated policies, where each policy corresponds to the target waypoint for a given storm and plane position, as shown in Figs. 2a and 2b for a single storm location and storm weights of $\beta = 2e^{1/2}$ and $\beta = 4e^{1/2}$, respectively. In these figures, the arrow stem corresponds to the plane location and the arrow head corresponds to the target waypoint. Thus, every arrow shows the optimal target waypoint for that particular location. As expected, the arrows divert more drastically around the storm for the case with the higher value of β.

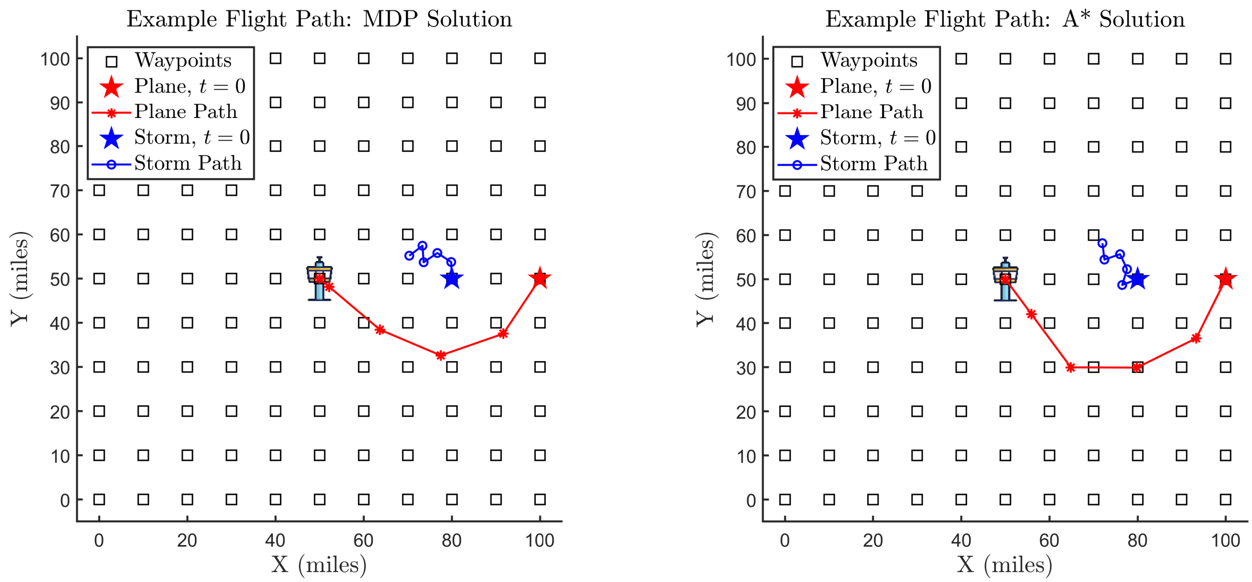


Fig. 3 Example flight paths from Experiment 1 for the (a) MDP solver and (b) A* solver. Markers along the paths represent plane and storm locations at intervals of $\Delta t = 5$ minutes.

B. A* Algorithm

The A* algorithm provides an online approach to the problem. It serves as an appropriate point of comparison to the offline MDP algorithm because A* is often used to solve for the shortest path in weighted-edge graph traversal problems. We implemented the A* algorithm with the same plane, storm, waypoints, and cost parameters; the formulation did not require an airport reward to be specified. In order to limit computational complexity, we only considered edges from any given waypoint to other points within a 5×5 gridpoint square centered on that given waypoint, rather than considering edges between all waypoints. This method is online in that the function recomputes the optimal A* path at every time-step with updated storm and plane positions. Although the complete path is computed, the plane is simply instructed to fly to the first waypoint in the sequence. Like the MDP approach, we adopt an interpolation scheme where we assume a discrete action space and continuous state space.

IV. Experiments

A. Experimental Setup

To evaluate the effectiveness of our aircraft rerouting algorithm, three different sets of experiments of $N = 10{,}000$ iterations each were run on both the MDP approach and A* search method. In each iteration, the plane was initialized at a random grid point along the edge of the grid world, while the storm was initialized at an interior grid point. The simulation then began where the storm moved according to the specified distribution of transition probabilities, and the plane followed its prescribed policy to reach the airport at the center of the grid world. Example simulations from experiments with the two solvers are given in Fig. 3. The evaluation metric used was average delay time, expressed as the percent increase of the plane's actual path length over the straight-line distance between the initial location of the plane and the airport. The standard deviation of this measurement was computed by bootstrap sampling over ten times the number of samples in the original experimental set in order to ensure convergence [10]. These values did not represent error bounds in the usual sense, because the data were not normally distributed and thus the lower bound yielded a negative number in several cases. Rather, they were reported to quantify the spread of the data and, in an approximate manner, the length of the tails of the distributions. Because of random initialization, only a fraction of all flights paths were impacted by the storm in a significant way. Thus, a separate set of statistics were reported for the set of samples where the percent delay exceeded 5%. The number of these samples was denoted N_D.

The parameters held constant in the experiments took the following values:

	Experiment 1	Experiment 2	Experiment 3
Fraction of Delayed Flights (N_D/N), MDP (%)	21.35	20.59	26.67
Fraction of Delayed Flights (N_D/N), A* (%)	19.82	28.08	33.79
Average Delay for All Flights, MDP (%)	3.44 ± 6.62	3.20 ± 6.18	5.62 ± 10.67
Average Delay for All Flights, A* (%)	4.32 ± 7.98	5.68 ± 9.54	8.15 ± 13.81
Average Delay for Delayed Flights, MDP (%)	13.84 ± 7.94	13.10 ± 7.59	19.52 ± 12.67
Average Delay for Delayed Flights, A* (%)	16.38 ± 11.62	16.00 ± 13.12	20.89 ± 17.77

Table 1 Statistics from the three experiments with the two solvers, computed over $N = 10,000$ samples. The number of delayed flights, N_D, is the number of flights that experienced delays of at least 5%. The error terms represent population standard deviations, estimated by bootstrapping.

- Simulation time-step: $\Delta t = 5$ min
- Storm standard deviation: $\sigma = 10$ mi
- Plane speed: $v = 3$ mi/min
- Storm speed: $u = 0.75$ mi/min
- Airport reward: $R_a = 1000$

With these parameters, the following three sets of experiments were carried out, in order to compare the performance of the two solvers for different storm conditions:

1) **Uniform storm transition:** The storm was initialized with uniform transitional probabilities along 45° intervals.
2) **Single random direction storm transition:** the storm selected a random direction along 45° intervals, and continued to move in that direction at constant velocity until the end of the simulation iteration.
3) **Doubled storm cost:** The weight of the storm in the cost function was increased from $\beta = 2e^{1/2}$ to $\beta = 4e^{1/2}$.

B. Experimental Results

The experimental results are summarized in Table 1 and Fig. 4. When comparing the two approaches, the average delay times were slightly lower when the plane followed the MDP policies across all three experiments. This result was found for averages computed over both N and N_D. This suggested that the MDP solver yielded shorter flight delays over all flights, as well as over flights with significant delays. Additionally, the standard deviations of the flight delays were uniformly larger for the A* solver. This suggests that the A* solver resulted in distributions of flight delays with longer tails, so that the worst-case flight delays of the A* solver were significantly larger than those of the MDP solver. These results are visible in the histograms shown in Fig. 4. Lastly, in two of the three experiments, the A* approach yielded a larger fraction of significantly delayed flights. Overall, these results showed that the MDP policy outperformed the A* algorithm by 2 to 3% in the mean. This, however, was partly due to the fact that the A* algorithm constrained the number of waypoints that could be chosen at each time-step, while the MDP policy allowed any of the M^2 possible waypoints to be selected. A broader action space for the A* algorithm would likely have decreased its average delays toward those of the MDP solver.

A broader action space for the A* algorithm, however, would also increase the computational complexity of the solution at each time-step. For the current experiments, the average runtimes per iteration were 0.002435 seconds and 0.2608 seconds for the MDP and A* solvers, respectively. Thus, the central difference between the performance of the MDP and A* solvers lies in the difference between offline and online solution methods. Both methods require a computationally intensive operation to calculate optimal actions. The difference between the two is that the MDP solver executes these computations before runtime, while the A* algorithm divides its computations into smaller problems that must be solved at runtime. The MDP requires $O(M^6)$ operations to compute the reward function, since all combinations of M^2 plane locations, M^2 storm locations, and M^2 waypoints must be considered. Similarly, the A* algorithm scales in complexity as $O(b^d)$, where b is the branching factor and d is the search depth. Depending on the heuristic used and restrictions on the accessible states and search depth, the A* algorithm could run more efficiently in an amortized sense than the MDP solver, but this cannot necessarily be guaranteed. The selection of one method over the other would thus depend on the size of the state and action spaces and the computing resources available for the task. For relatively small M, such as the value considered in this work, the MDP solver would likely be a better choice, since it computes an

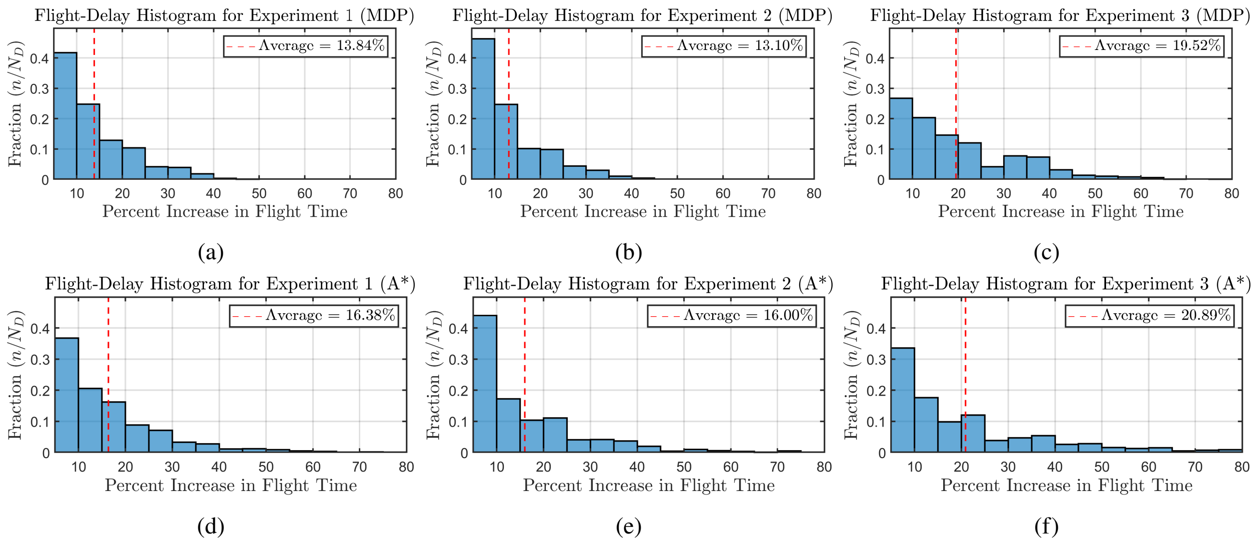


Fig. 4 Histograms of the number of delayed flights (n), normalized by the total number of delayed flights in each experiment (N_D) and binned by the percent increase in flight time, for the three experiments with the MDP solver (a-c) and the A* solver (d-f). Flight delays of less than 5% were not included in these analyses.

optimal policy, has a runtime that is independent of M, and does not require the heuristics and action-space restrictions of the A* implementation. For larger M, computing an optimal policy using value iteration would become infeasible, and approximate solutions to the MDP-based problem formulation would have to be employed. At this point, the A* solver's acceptable performance with reduced dimensionality would become more attractive.

V. Conclusions

In this work, we presented a formulation to reroute aircraft around severe weather events. An offline MDP solver was used to determine the optimal policy for every combination of plane and storm positions, and was implemented in the simulation as a look-up table. The performance of the offline MDP solver was then compared to the online A* algorithm. The results of experiments conducted using these solvers demonstrated that the MDP approach resulted in fewer and less significant flight delays in comparison to the A* approach. However, the A* algorithm still performed reasonably well, and would also be effective for dynamic aircraft routing in situations where a much larger grid dimension is needed.

Additional work can be done to improve the MDP algorithm. One avenue of particular interest is adding an additional component of storm prediction. The current solver selects the optimal policy only based on the storm and plane position. However, the solver can be upgraded to account for the storm's heading via a belief state over the storm heading. This avenue could lead to more optimal path planning where the plane avoids diverting in the same direction as the storm heading. Motivation for adding this belief state stem from an incident in 2013 where air traffic controllers rerouted planes in the same direction as an evolving storm front, resulting in 69 diverted aircraft, and 55 airborne holdings, with almost 600 departure and arrival cancellation [1]. Other avenues of interest include adding other airplanes into the simulation and avoiding mid-air collisions, enforcing airplanes to approach the airport from a particular vector, and implementing multiple runways These future areas of research could open up a more robust and realistic autonomous air traffic controller that could safely route planes in the presence of adverse weather.

Author Contributions

C.S. implemented the A* algorithm, conducted the experiments, and set up the grid-world class. P.W. implemented the MDP algorithm, set up the plane class, and wrote visualization scripts for the simulations. N.J.W. implemented the storm class and cost functions, set up the experiment scripts, and conducted the statistical analysis. All the authors contributed equally to the overall problem formulation and algorithm design, and compiled the text and figures.

References

[1] Administration, F. A., "FAQ: Weather Delay," , 8 2017. URL `https://www.faa.gov/nextgen/programs/weather/faq/`.

[2] Balaban, E., Roychoudhury, I., Spirkovska, L., Sankararaman, S., Kulkarni, C. S., and Arnon, T., "Dynamic Routing of Aircraft in the Presence of Adverse Weather Using a POMDP Framework," *17th AIAA Aviation Technology, Integration, and Operations Conference*, American Institute of Aeronautics and Astronautics, Denver, CO, United States, 2017. https://doi.org/10.2514/6.2017-3429, URL https://arc.aiaa.org/doi/10.2514/6.2017-3429.

[3] Tompa, R. E., Kochenderfer, M. J., Cole, R., and Kuchar, J. K., "Optimal aircraft rerouting during commercial space launches," *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, 2015, pp. 9B1–1–9B1–9. https://doi.org/10.1109/DASC.2015.7311486.

[4] Tompa, R. E., and Kochenderfer, M. J., "Optimal Aircraft Rerouting during Space Launches using Adaptive Spatial Discretization," *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018, pp. 1–7. https://doi.org/10.1109/DASC.2018.8569888.

[5] Ćurić, M., "Numerical modeling of thunderstorm," *Theoretical and Applied Climatology*, Vol. 40, No. 4, 1989, pp. 227–235. https://doi.org/10.1007/BF00865973, URL https://doi.org/10.1007/BF00865973.

[6] Weisman, M. L., and Klemp, J. B., "Characteristics of {Isolated} {Convective} {Storms}," *Mesoscale Meteorology and Forecasting*, edited by P. S. Ray, American Meteorological Society, Boston, MA, 1986, pp. 331–358. https://doi.org/10.1007/978-1-935704-20-1{_}15, URL https://doi.org/10.1007/978-1-935704-20-1_15.

[7] Kopardekar, P., Rios, J., Prevot, T., Johnson, M., Jung, J., and Iii, J. E. R., "Unmanned Aircraft System Traffic Management (UTM) Concept of Operations," *16th AIAA Aviation Technology, Integration, and Operations Conference*, 2016. https://doi.org/10.2514/6.2016-3292, URL http://arc.aiaa.org.

[8] Foina, A. G., Krainer, C., and Sengupta, R., "An Unmanned Aerial Traffic Management solution for cities using an air parcel model," *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, Institute of Electrical and Electronics Engineers Inc., 2015, pp. 1295–1300. https://doi.org/10.1109/ICUAS.2015.7152423.

[9] Chadès, I., Chapron, G., Cros, M.-J., Garcia, F., and Sabbadin, R., "MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems," *Ecography*, Vol. 37, No. 9, 2014, pp. 916–920. https://doi.org/10.1111/ecog.00888, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ecog.00888.

[10] Efron, B., "Bootstrap Methods: Another Look at the Jackknike," *The Annals of Statistics*, Vol. 7, No. 1, 1979, pp. 1–26.