Fifteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2023)

Teaching Artificial Intelligence Good Air Traffic Flow Management

Christine Taylor, Erik Vargo, Tyler Manderfield, and Simon Heitin
Center for Advanced Aviation System Development
The MITRE Corporation
McLean, VA, USA
ctaylor@mitre.org, evargo@mitre.org, jmanderfield@mitre.org, sheitin@mitre.org

***Abstract*—Air Traffic Flow Managers are continually faced with the decision of when and how to respond to predictions of future constraints. The promise of Artificial Intelligence, and specifically reinforcement learning, to provide decision support in this domain stems from the ability to systematically evaluate a sequence of potential actions, or strategy, across a range of uncertain futures. As decision support for human traffic managers, the generated recommendations must embody characteristics of a good management strategy; doing so requires introducing such notions to the algorithm. In this paper, we propose to induce *stability* into the strategy by dynamically constraining the design space to promote consistency across decisions. We further evaluate the impact of adding a performance improvement threshold that must be overcome to accept a new strategy recommendation. The combination of search constraints and threshold values is evaluated against the agent's reward function in addition to measures proposed to capture the stability of the strategy. The results demonstrate that strategy stability can be improved without unduly impacting performance.**

Keywords- artificial intelligence; reinforcement learning; traffic flow management; sequential decision making; decision support

I. INTRODUCTION

The goal of strategic Air Traffic Flow Management (ATFM or simply, TFM) is to balance demand with resource capacity in a safe and efficient manner. To mitigate situations where demand is predicted to exceed capacity, a TFM strategy – which comprise a sequence of Traffic Management Initiatives (TMIs) – is used to modulate future demand. The fundamental challenge in designing TMI strategies arises from the significant forecast uncertainties present at the time decisions are made, creating a dilemma of when and how to respond. As such, in today's operation in the National Airspace System (NAS), TFM strategies are developed through a collaborative multi-stakeholder decision making process which is heavily reliant on past experiences and events.

The need for sequential decision-making approaches (i.e., methods that account for the dynamics of uncertainty in the timing of decisions) to provide decision support in this domain has long been recognized. Earlier efforts proposed a variety of approaches to tackle components of this challenge. For example, mixed-integer programming algorithms for GDP planning [1-4] demonstrated success at overcoming the large TMI design space, albeit under constructed, rather than derived, constraint scenarios. Taking an operational view of the decision process, [5] developed a TFM strategy defined decision tree that could be used to assist in the sequential decision-making. Alternatively, research into deriving constraint scenarios demonstrated the importance of capturing the physics phenomena [6] in the definition of constraint scenarios but did not take the additional step of defining the mechanics of how to connect these scenarios to the TFM decision-making environment. To bridge this gap, [7] employed a Genetic Algorithm to optimize a multi-time course TFM strategy against an ensemble of derived forecast scenarios; however, computational limitations precluded real-time decision support.

Recently, the resurgence of Artificial Intelligence (AI), and specifically Deep Reinforcement Learning (DRL) techniques, have shown promise for these overcoming computational issues. In the AI domain, an agent refers to automation that identifies and selects an action(s) to implement with the goal of maximizing a reward. Tien et. al [8] developed an AI agent to sequence taxiing aircraft at a hypothetical airport. Brittain and Wei [9] trained an agent controller to maintain aircraft separation in congested airspace, albeit assuming perfect information. The approach developed in [10] incorporates uncertainty into the conflict avoidance and separation assurance problem by assuming a normal distribution on the resulting state of an aircraft after executing the agent-prescribed action. Tran et. al [11] built upon this work to learn controller-specific resolutions with the goal of gaining greater operational acceptance.

Research into developing AI agents to generate TMI strategies is more limited. Xie et al. [12] developed a DRL-based agent to generate tactical ground delay actions targeting specific airports, with the goal of alleviating congestion in Air Traffic Control (ATC) sectors; however, given the tactical nature of the problem, forecast uncertainty was not considered. Alternatively, Jones et al. [13] examined the performance of several different search techniques for selecting TMI recommendations from a set of pre-defined combinations, where uncertainty was captured by sampling across forecast future. Similarly, our earlier work [14] employed a Monte Carlo Tree Search (MCTS) to generate TMI recommendations for the next hour based on a translated ensemble weather forecast. However, both this and the previous effort did not employ DRL, essentially requiring these agents to learn each decision anew.

In our previous work [15], we demonstrated the ability of an Expert Iteration (ExIt) algorithm to generate TMI recommendations under forecast uncertainty in a real-time context. Importantly, this work leverages DRL to learn a policy network (PNet) which, when coupled with a statistical model to improve the forecast error, demonstrated skill at overcoming performance degradations due to uncertainty.

While our previous work demonstrates the viability of this technology, the specific TMI recommendations were not necessarily acceptable within the context of decision support to a human Traffic Manager (TM). The agent was permitted to make a recommendation every hour and often did. While updated recommendations are not inherently undesirable, the changes, if inconsistent from one decision to the next, would result in an increase in workload for TMs, a decrease in predictability for flight operators, and thus an overall reduction in trust and adoption for a future decision support capability.

The objective of this paper is to identify and evaluate mechanisms to improve the TFM agent's TFM strategy recommendations. In discussions with Subject Matter Experts (SMEs), a dominant theme has been the desire for the agent to generate a stable TFM strategy (i.e., that the recommendations should gracefully evolve over the planning horizon). While the agent is not capable of understanding grace, it does abide by constraints. As such, we propose to dynamically limit the action space based on the current recommendation sequence; we define two such constraint sets and compare these against the baseline, unconstrained search. A second consideration raised was the desire for a user-defined threshold to limit when the agent alerts a TM of a new TMI recommendation. While we envision that this threshold in performance improvement would vary, both by TM and likely by situation, we propose to evaluate the impact of *waiting* – ignoring the TMI recommendation and remaining with the current plan – for a small threshold level. Each of the three constraint sets (including the baseline) are considered with and without a threshold imposed on enacting the recommendation. To evaluate the results, we consider the performance of the TFM strategy, as defined by the agent's reward function, as well as additional measures that capture strategy stability.

The remainder of the paper is organized as follows. Section II details the environment for our case study, where a brief description of the scenarios, TMIs and reward function are presented. Section III provides an overview of the TFM agent decision framework. The specifics of the proposed experiment, including the additional stability measures considered are provided in Section IV. The results and discussion are presented in Section V and Section VI summarizes the conclusions.

II. Case Study

Our case study focuses on designing TMIs to manage arrival demand into Atlanta Hartsfield Jackson International Airport (ATL) under uncertain weather and potentially degraded capacity conditions. As shown on the right side of Figure 1, our representation abstracts the arrival routes into a four corner-post configuration common at ATL. The four corner-posts, termed *fixes* for the remainder of the paper, are labeled as Northwest

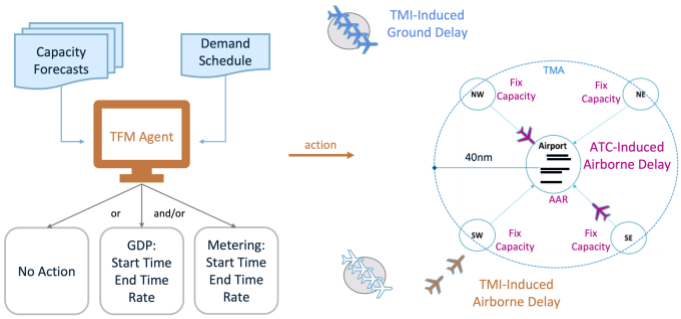

Figure 1. Case Study of Managing Arrivals into ATL under Forecast Uncertainty

(NW), Northeast (NE), Southwest (SW), and Southeast (SE) and are positioned 40 NM from the airport, which is shown in the center of the circle.

The left side of Figure 1 highlights the information provided to the agent and the TMIs that the agent can take, which will be described later in this section. Note that while the agent plans TMI actions based on forecasted capacities, the performance is assessed against observed capacities for the scenario.

A. *Scenario Generation*

Scenario data was generated for the period between 1 January 2015, and 1 January 2020. Each scenario corresponds to a historical day containing a time history of demand at each of the five resources (the four fixes and the airport), and the capacities – both derived from forecast and observation – for these resources.

Demand data is derived from the first filed flight plan for each flight arriving at ATL. As described in [14], the flight plans were mapped to one of the four fixes. The average transit time between each fix and the airport was derived and used to estimate the expected time of arrival to each fix, based on the expected time enroute (ETE) for each flight.

The airport capacity is defined as the Airport Arrival Rate (AAR) associated with the most common configuration between 2015-2019, namely 26R 27L 28 | 26L 27R. Reference [16] provides the AARs for each of the four Meteorological Conditions (MC): Visual MC (VMC), Low VMC (LVMC), Instrument MC (IMC), and Low IMC (LIMC), where the published hourly rates were divided by four and rounded down to provide the 15-minute values shown in Table 1.

To compute the applicable observed MC rate, we used Automated Surface Observing System (ASOS) and Meteorological Terminal Air Report (METAR) data to identify the actual ceiling and visibility for each 15-minute period and recorded the associated AAR for the airport at that time [17].

Nominal fix capacities were estimated as the 95% value of historical arrival throughput in each quadrant, as shown in Table 2. To compute the actual weather-impacted capacities,

Table 1. Meteorological Conditions for ATL Case Scenario

Category	**Visual MC (VMC)**	**Low VMC (LVMC)**	**Instrument MC (IMC)**	**Low IMC (LIMC)**
Rate per 15 min	33	31	27	24

we leveraged the Corridor Integrated Weather System (CIWS) nowcast to provide measurements of Vertically Integrated Liquid (VIL) greater than or equal to 3 mm of surface accumulation (VIL3+) in the 80 NM area surrounding the airport for each 15-minute period. A SME vetted modification [15] to the relationship provided in [18] was used to compute the fraction of nominal capacity available.

Table 2. Nominal Fix Capacities per 15-minute time bin

Fix	**NW**	**NE**	**SW**	**SE**
Capacity	10	10	7	7

To compute the predicted resource capacities which will be used by the agent to select TMI actions, we leverage the Short-Range Ensemble Forecast (SREF), which consists of 26 deterministic trajectories of weather variables at hour-long intervals. As stated in [19], each member of the ensemble is equally likely to occur and, together, the ensemble members span the space of future outcomes. Furthermore, a new SREF is issued every 6 hours.

Using the SREF data, we compute the applicable MC to obtain a prediction of AAR and the reflectivity to estimate VIL3+ coverage, as described in [19]. The resulting predicted capacity ensemble contains 26 members, where each member contains the 15-minute integer capacities (e.g., hourly capacities divided by 4) for each of the five resources.

B. Traffic Management Intitiatives

Two types of TMIs are considered in this case study: GDPs and metering. A Ground Delay Program (GDP) is a strategic TMI that delays flights on the ground prior to departure. The GDP is defined by four parameters:

- Rate: The maximum quarterly arrival rate for flights.
- Scope: The scope defines the set of origin airports whose departures are subject to delays by the GDP.
- Start time: The start time of the GDP expressed in local time at the destination airport.
- Duration: The duration of the GDP.

For the case study in this paper, the scope was set to include all departures.

The GDP implementation was based on the logic of Flight Schedule Monitor (FSM) [20] and incorporates the option to cancel or revise an existing GDP (i.e., alter the GDP parameters). Arrival slots are assigned using a ration-by-schedule logic where controlled arrival times (CTAs) and controlled departure times (CTDs) are computed based on the assigned arrival slot times and the flights' ETEs.

In the case of revisions, delay is released (to the extent possible) from flights no longer included in the revised GDP and re-assigned based on the new slots, where flights that were impacted by the previous GDP have precedence over flights that were not included. Additionally, flights can be exempt for several reasons (scope, departure time), and exempt flights are assigned slots before any other groups. In contrast to non-exempt flights, exempt flights take up slots and are assigned CTDs and CTAs but are not delayed.

The metering TMI is intended to represent tactical, coordinated air delay assigned to flights prior to arrival in the terminal airspace. Each fix can have a separate "metering TMI" that is defined by:

- Rate: The permissible arrival rate per 15 minutes. This rate is translated into a new time of arrival at each corner-post fix.
- Start time: The start time of the rate restriction.
- Duration: The duration of the metering program in minutes.

C. Reward Function

The total delay accrued for each flight is the sum of the TMI Ground Delay (d^g), TMI Air Delay (d^m), and ATC-induced Delay (d^a). The TMI Ground Delay for a flight is calculated as the difference between the CTD and the original estimated time of departure (OETD). TMI Air Delay is measured as the difference between the scheduled fix arrival time and the assigned fix arrival time resulting from the metering TMI. ATC-induced delay is the total queuing delay imposed by the simulation, capturing both queues at the fix and at the airport. Details on the queuing simulation that calculates ATC-induced delay can be found in [15].

The reward function is defined to minimize the *delay impact* for the entire scenario, where delay impact represents a non-linear aggregation of the three components of delay that aims to reflect the relative "pain" caused. We therefore considered it unitless. Using Subject Matter Expert (SME) guidance, the delay impact for each type of delay was defined as a piece-wise linear function to capture not only the difference between sources of delay but how the duration of the delay changes the impact. Figure 2 displays these relationships.

Viewing Figure 2, we see that for flight-specific delays that are less than two minutes, ATC-induced delay (red line) returns the least delay impact; however, as ATC delays increase, delay impact grows at the fastest rate to reflect the increased workload associated with absorbing different amounts of delay. For delays greater than 30 minutes, a large penalty is assigned to represent the disruption associated with a potential diversion. For flight-specific delays between two and ten minutes, metering delay (orange line), induces the least delay impact and increases more slowly than ATC delay, but still assigns a large penalty for delays over 30 minutes. GDP delay is the least impactful for delays over 10 minutes and does not have a large penalty assigned at higher delay values as delay is taken on the ground. The reader is referred to [15] for the specific equations.

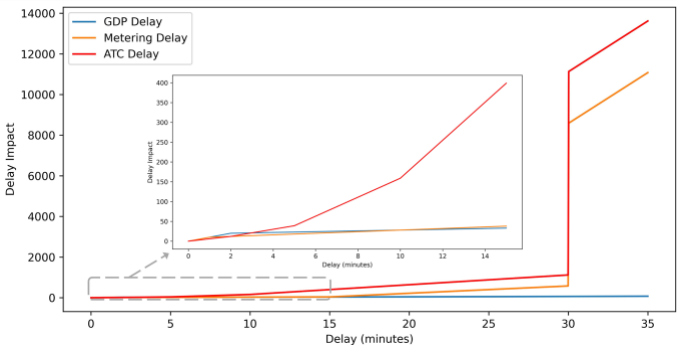

Figure 2. Delay Impact vs Delay Minutes

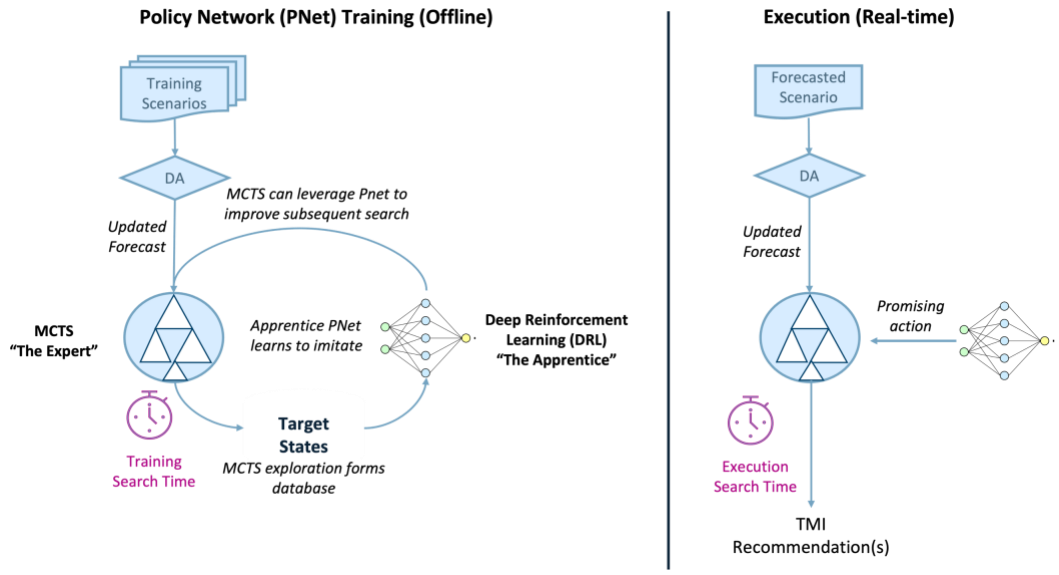

Figure 3. Overview of Expert Iteration Algorithm

III. TFM AGENT

Figure 3 provides an overview of the TFM agent developed in [15]. The left side shows the Expert Iteration (ExIt) algorithm [22] which uses approximately five years of historical training scenarios to encode the relationship between the *state* – represented as the set of expected demand and capacity scenarios, along with observed system behavior up to the current time – with the TFM strategies that are most likely to improve the reward. The right side of Figure 3 depicts the real-time evaluation process for each hour of the new forecasted scenario, where reliance on the offline policy network (PNet) improves the agent's ability to quickly identify which actions are likely to yield better outcomes.

A key challenge noted in [14] was the impact of capacity prediction error on generating TMIs that respond to the constraints. To address this issue, a Data Assimilation (DA) algorithm was developed to update the probabilities associated with each capacity ensemble member based on their agreement with the previously observed capacity values over the past several hours.

Central to both the development of PNet and the real-time execution is the MCTS algorithm, which is briefly described below. For a thorough development of each of the agent's algorithmic components, the reader is referred to [15].

The agent uses MCTS to select the *best* action – i.e., the action that the agent estimates will achieve the lowest future expected delay impact relative to the predicted capacities – at the current (hourly) decision time t. During the duration allotted, MCTS builds a tree that estimates the optimal *adaptive* TMI policy with respect to the possible future capacity trajectories as derived from the capacity ensemble. The policy is *adaptive* over the target planning horizon in the sense that the best action at time t considers future *contingencies* – downstream TMI actions that could be triggered by future observed capacities. Note, however, that the time t tree is used to select only the time t action, and a new tree is built to select the action at each subsequent decision time. The agent's action at time t results in an updated system state, which is used to initialize the topmost "root" node of the time $t+1$ tree.

Our MCTS algorithm is initialized by defining a root node N_0 at depth $d = 0$. The root node reflects the current state of the TFM environment at time t, including the impact of all actions taken up to time t on each flight's position in the queueing model, and by extension projected future demand at each of the five capacitated resources. In general, we let S_N denote the set of SREF members that are consistent in their capacities along the path in the tree from the root N_0 to node N. It follows that S_{N_0} contains all 26 members since at depth $d = 0$ no capacities have been branched on to distinguish one member from another.

Before optimizing over TMI actions, we first construct a "baseline" tree from the root downward that captures the space of possible futures when no TMI actions are taken (but assumes actions taken prior to time t are in place). The possible futures are enumerated in tree form as follows: From the root node, first identify which members in the capacity ensemble have the same capacity values over the next hour (though the values may vary over time) and place these into the same set S at depth $d = 1$, and associated with each set we create a child node of the root, where S_C is the set of ensemble members associated with a child node C. Note that the collection of all S_C for child nodes C of any parent node N partitions the parent's consistent SREF set S_N. For notational convenience, we also define $N_0 = \text{Parent}(C)$ for all children C of N_0. For the baseline tree construction, the parent-to-child transitions assume that no TMI action is taken, and we represent this no-TMI action as $\tilde{a}$. We apply this branching process recursively to each child to create a complete tree. We terminate the branching process when the child node depth reaches a predefined target value of $d_{max} = 5$, or the child node represents a terminating state of the simulation in which all flights have landed – whichever comes first. By limiting the depth of the tree in this way, we ensure that the MCTS algorithm will focus its action search on more immediate actions (here, over the next 5 hours) where the future – and hence action impact – is more certain.

The next step is to initialize a value function at all nodes in the baseline tree. This process begins at the leaf nodes of the baseline tree and propagates upward to the root. If a given leaf node N does not represent a terminating state of the simulation, we estimate the value $V_{N,\tilde{a}}$ of action $\tilde{a}$ at the node by performing a *rollout* from N – otherwise we set $V_{N,\tilde{a}} = 0$. In our MCTS implementation, a rollout simulates the current node state to the completion of the simulation, wherein all future actions default to $\tilde{a}$. The rollout is executed for each ensemble member in $s \in S_N$ and their resulting future rewards are combined in a weighted average, proportional to their respective data assimilated probabilities $p(s)$ to obtain $V_{N,\tilde{a}}$. In general, the optimal action at node N is given by

$$a_N^* = \underset{a}{\operatorname{argmax}} V_{N,a} \qquad 1$$

where the argmax is taken over all actions a explored at node N. In the baseline tree, $a_N^* = \tilde{a}$ since no other actions have yet been explored. The corresponding optimal value at node N is denoted $V_N = V_{N,a_N^*}$. Next, we use dynamic programming to recursively update parent node values for action $\tilde{a}$:

$$V_{N,\tilde{a}} = \sum_{C \in \text{Children}(N,\tilde{a})} w_C \cdot \left(r_{C,\tilde{a}} + V_{C,\tilde{a}}\right) \qquad 2$$

Here w_C denotes the DA-weighted SREF probabilities associated with S_C, as defined in Equation 3.

$$w_C = \frac{\sum_{s \in S_C} p(s)}{\sum_{s \in S_N} p(s)} \qquad 3$$

Just like with the leaf nodes, we let $a_N^* = \tilde{a}$ represent the optimal action at each subsequent parent node N until the recursive process terminates at the root. A notional illustration of the resulting baseline tree is depicted in Figure 4 for $d_{max} = 2$, where dashed lines represent rollouts from leaf nodes. Note that all transitions from parent to child involve the "no TMI" action $\tilde{a}$.

Once the baseline tree is constructed, the MCTS algorithm explores new actions to improve upon the baseline policy, where our specific MCTS implementation is inspired by the Combinatorial Multi-armed Bandit problem [22] to manage the combinatorically large action space.

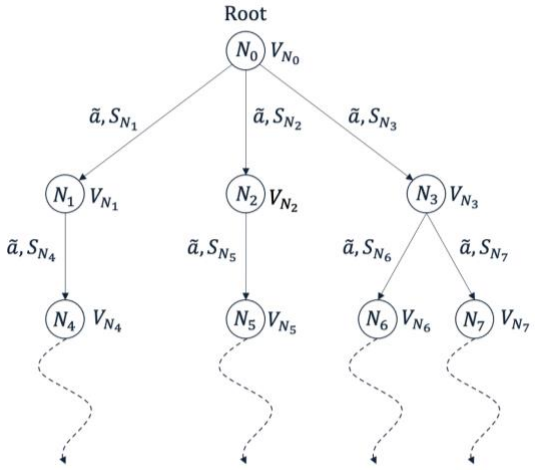

Figure 4. Notional Depiction of Baseline MCTS Tree

IV. Stability Experiment

This paper evaluates how changes to the TMI search and recommendation process impact the structure of the resultant TFM strategy. Based on SME input, we propose two different mechanisms to induce stability in the agent's TMI recommendations. The first constrains the search space at decision nodes during MCTS, where two different sets of constraint rules are defined. The second imposes a threshold on the expected performance improvement before a TMI recommendation is implemented at the current decision time. To evaluate the impact of these changes, we define measures that aim to reflect the stability of the TFM strategy from the perspectives of both a TM and a flight operator.

A. Search Space Constraints

The MCTS action selection and evaluation process described in Section IV provides the agent maximum flexibility to address the range of forecast futures; however, it can result in TMI recommendations that do not reflect a good operational strategy. For example, the agent could select a GDP at a decision time only to cancel this action at the next and potentially re-implement at the subsequent decision time. Providing such a strategy to a TM would not only be unhelpful but would likely degrade trust in the automation. As such, we seek to reflect the desire for an adaptive, yet consistent plan by requiring the agent to limit subsequent actions based on the history of decisions at both the root node and at downstream contingency planning nodes. While the TMI recommendation is generated solely from the root node, constraining downstream nodes requires the agent to evaluate the goodness of the root node action in the context of a stable plan.

We consider three constraint cases in this paper. The first, Case 1, imposes no constraints on the search space (i.e., baseline case). Case 2 restricts the search space as follows:

- If a GDP action is selected, the rate change is limited to +/- 25% of the current rate
- If a GDP action ends or is canceled, no new GDP action may be initiated for four hours
- If a metering action is selected for a specific fix, the rate change is limited +/- 15% of the current rate
- Once a metering action ends, no new metering action at the same fix may be initiated for 2 hours.

Case 3 assumes the same restrictions as Case 2 and adds an additional constraint on GDP actions that prohibits any changes within two hours of when it was implemented or last changed.

B. Improvement Threshold

The second modification addresses SME concerns regarding the frequency of TMI recommendation alerts, especially in cases where the recommendation does not meaningfully change the expected performance. While we expect that the quantification of a meaningful change will vary by TM and scenario, we impose a performance improvement threshold to determine whether a recommended decision is implemented.

To motivate the selection of a performance threshold, Figure 5 presents a histogram of samples, representing decision

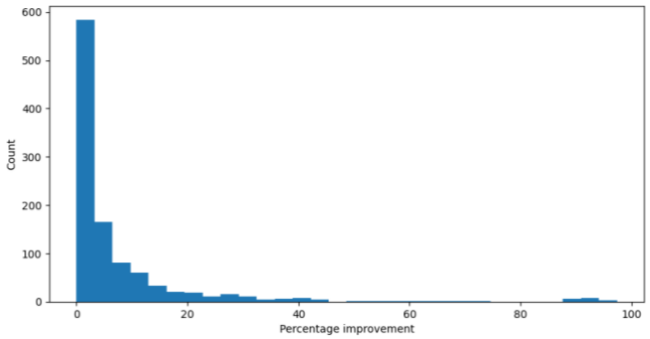

Figure 5. Histogram of Percent Improvement of Expected Reward due to Implementing a TMI Recommendation verses Waiting

points where the agent recommended a new action. Each sample in the histogram corresponds to an action taken during the 92 validation days without any constraints (Case 1), where the percent of expected performance improvement associated with taking the action is calculated. Viewing Figure 5, we note that over half (55%) of the recommendations only provide a marginal (less than 3.2%) improvement in expected reward; however, the distribution has a long tail, corresponding to decision points where not taking a recommendation at that time (i.e., waiting) is likely to have a significant impact on future expected reward.

C. *Measuring TMI Stability*

To quantify the impact of the proposed changes on TFM strategy stability, we consider several measures to reflect this behavior from the perspective of the TM or the flight operator. The first measure computes the number of individual TMI recommendations generated, where GDP revisions are included within this count, but TMI extensions (i.e., when the agent issues a recommendation that extends the current TMI without any change to the rate) are exempt from this calculation.

The second measure focuses on the variability in TMI rate. It is important to note that in this research, the agent can only assign a single rate to a TMI recommendation and therefore it may be desirable to change the rate at a future time, in response to changes in the forecast. However, if the agent's recommendation strategy repeatedly alternates the rate up and down, this will have a negative impact on both TMs and flight operators. As such, we compute the number of inflection points in the rate profile for each TMI over the course of the day.

Our final measure aims to reflect GDP stability from the perspective of a flight operator. Specifically, we compute the number of flights with multiple EDCTs assigned and focus on the additional ground delay accrued during revisions. The motivation here is that while the initial GDP may disrupt the original plan, subsequent revisions reduce predictability. We further categorize the magnitude of this additional delay into 3 bins: less than 15 minutes, between 15 and 30 min and greater than 30 minutes of additional delay, to reflect the varying degrees of disruption produced by GDP revisions.

V. RESULTS

A. *Experimental Design*

To evaluate the performance of the TFM agent, we generated TFM strategies for three non-consecutive months: June, October, and December 2019. For each scenario day, the agent was provided with the capacity ensemble and had an hour to generate the recommended action for the upcoming hour. For the two TMIs described in Section II.B, we permitted the agent to select among the TMI parameter values listed in Table 3.

As shown in Table 3, the TMI start time (in minutes)

Table 3. TMI Parameter Options

TMIs	**Start Time (min)**	**Duration (min)**	**Rate (per 15 min)**
GDP	{60, 120, 180, 240}	{120, 240, 360, 480}	{4, 8, 12, …, 32}
Metering	{30, 60, 90}	{60, 90, 120, 180, 240}	NW/E: {1, 2, …9} SW/E: {1, 2, …9}

specifies the offset between the decision time and the start of the program. For the GDP, this offset can dictate which flights will receive ground delay, as flights within 30 minutes of their original departure time will be exempt. The duration (in minutes) specifies how long the TMI will be in effect. For GDP actions, the rate specifies the number of aircraft that can land at the airport in a 15-minute bin. For metering actions, the rate specifies the number of flights within each 15-minute bin that can cross the corner post fix, where the difference between the two upper limit values distinguishes between the nominal capacity of 10 flights per 15-minute bin associated with the two northern fixes and 7 flights per 15-minute bin for the southern fixes. In addition to these choices, each agent can opt to take no action, cancel, or revise an existing action. Note that while only two TMI types are defined and only a handful of parameter values are considered for each, the resulting design space has more than 40 billion discrete action choices at each time step and for each decision depth of the tree.

B. *Evaluation of 92 Day Validation Set*

Figure 6 shows the total delay impact incurred by each constraint case described in Section IV for each of the 92 days when no threshold is applied. Viewing Figure 6, we note that June 8th is the most impactful day and results in the largest performance variation between the three cases, where Case 2 incurs the highest delay impact and Case 3 yields the lowest delay impact. This may seem counterintuitive – the addition of constraints cannot improve the optimality of a solution – however, MCTS has no guarantee of optimality and given the combinatorially large search space, the addition of constraints is likely to assist the agent during its search.

Viewing Figure 7, we see that the 1% threshold does not noticeably change the overall delay impact across the 92 days, except for June 8th, where all three cases roughly achieve the same delay impact, with Case 1 performing the best and Case 3 performing the worst. This is not surprising – referring to Figure

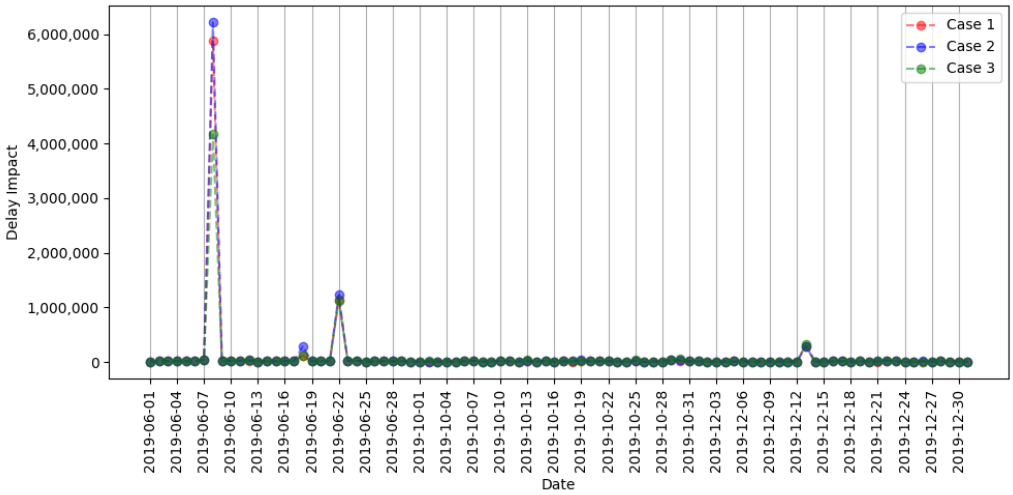

Figure 6. Comparison of Constraint Case Delay Impacts without Threshold over 92 Validation Days

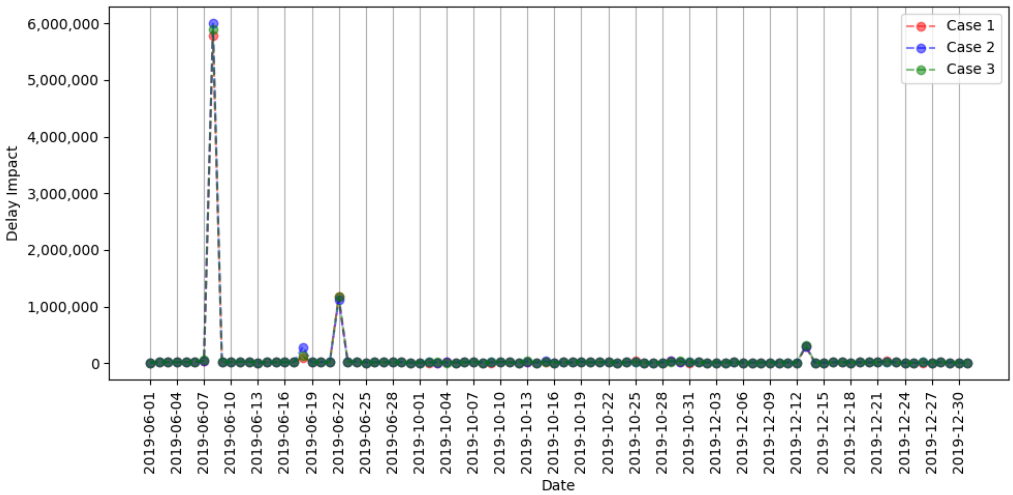

Figure 7. Comparison of Constraint Case Delay Impacts with 1% Threshold over 92 Validation Days

5, we note the long tail of performance improvement loss due to waiting, where presumably larger differences correspond to days with larger capacity reductions on which delaying action can significantly degrade expected future performance.

To gauge the impact on stability verses total delay created by the proposed changes, we examine the difference between Case 2 and Case 3, as compared to Case 1 in the number of GDPs (Figure 8) and the number of metering actions (Figure 9) compared to the total delay, where total delay is the sum of the three delay components without delay impact weighting. The left side of Figure 8 shows the results without implementation of a threshold for days where a GDP was issued. As the points represent the difference between Case 2 and Case 1 (pink) or Case and Case 1 (blue), we can interpret the plots as follows. Points in the upper right quadrant represent days when the constraint cases resulted in higher total delay and more TMI actions than the baseline, whereas the bottom left quadrant captures days where the constraints reduced both. Points in the upper left quadrant correspond to days where the constraints reduced the number of TMIs, but at the expense of additional delay, whereas the bottom right quadrant corresponds to days where the constraints resulted in more TMIs but lower overall total delay.

When no threshold is applied, we see that, in general, Case 2 not only performs worse than Case 1 most of the time, but worse than Case 3 as well, implying that simply restricting the rate ranges may be insufficient for inducing TMI stability and may lead to higher delay costs as well. As expected, Case 3 tends to reduce the number of GDPs issued, but higher delays may be incurred since the specific GDP cannot be adapted for 2 hours, despite any changes in the constraint forecast.

Comparing the results when a 1% performance improvement is implemented, we see that this restriction reduces the variance in total delay between the constraint cases and the baseline. Case 3 continues to outperform Case 2, reducing both the number of GDPs issued and the total delay incurred. However, we note two days when Case 3 resulted in both an additional GDP and higher total delay, as compared to Case 1, indicating the potential for performance degradation due to waiting on high-impact days.

Figure 9 displays the difference in the number of metering actions verses total delay generated for Case 2 and Case 3, as compared to Case 1 when no threshold is applied. Recall that

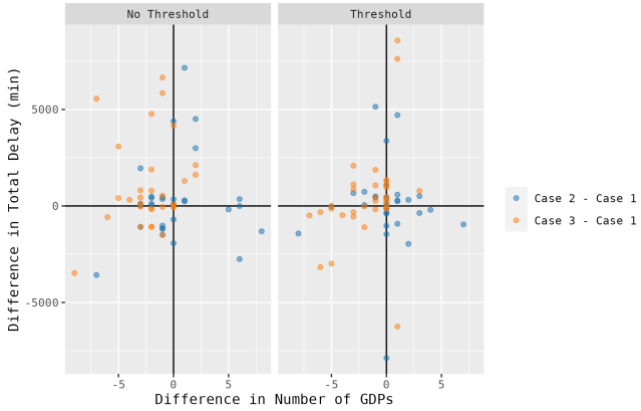

Figure 8. Difference in Number of GDP actions for each constraint case compared to baseline verses total delay

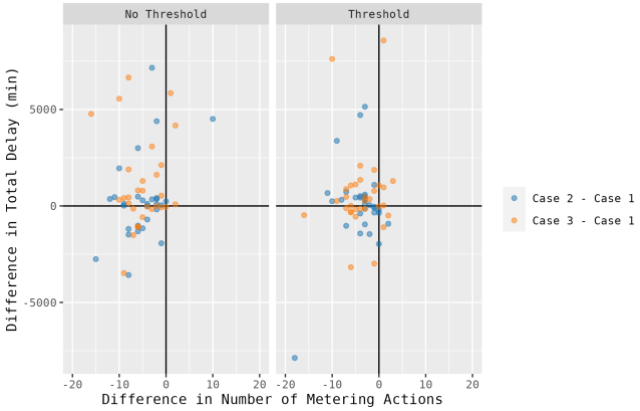

Figure 9. Difference in Number of Metering actions for each constraint case compared to baseline verses total delay

both Case 2 and Case 3 impose the same constraints on metering actions and as a result both significantly decrease the number of recommendations generated by the agent. When thresholding is applied, we notice that Case 2 continues to report a similar decrease in metering actions as well as a reduction in total delay, as compared to Case 1. Case 3, however, has a slight increase in relative metering actions, as compared to the baseline, which may result from a greater need to manage traffic through metering, given the GDP restrictions imposed.

We next compare the number of inflection points in the GDP rate profile for days when a GDP was issued, as shown in Table 4. For each constraint case we distinguish between results when the threshold was not applied (NT) and when it was implemented (TH). Viewing Table 4, we see that for Case 1 and Case 2, the implementation of the threshold decreases the number of inflection points in the resultant GDP strategy, indicating that "waiting" can reduce the tendency of the agent to overoptimize the solution in pursuit of improvement in delay impact. However, Case 3, either with or without thresholding has the greatest impact on the number of inflection points, as expected, since rate changes are explicitly disallowed for 2 hours. Thus, from this perspective, Case 3 provides the greatest stability in the GDP strategy.

C. June 8th 2019

Our final analysis focuses on the highest delay impact day, June 8th. Figure 10 presents the number of EDCT revisions issued for each TFM strategy developed and the corresponding additional ground delay associated with the revision, as categorized into three delay ranges: minor (less than 15 minutes), moderate (greater than 15 minutes but less than 30 minutes), and significant (greater than 30 minutes). Note that a single flight may be subject to more than one revision and each revision is represented as a sample in Figure 10.

The left side of Figure 10 compares all three cases when no threshold is applied, where we see that Case 2 results in the greatest number of revisions and more instances of moderate and significant additional ground delay. Case 3 incurs the least number of revisions, however there is an increase in moderate and significant revision delays, as compared to Case 1.

The right side of Figure 10 compares the three cases when thresholding is applied. Viewing Figure 10 we note that both Case 1 and Case 2 result in more EDCT revisions when thresholding is applied than without and that there is an increase in the number of revisions that incur significant delay. However, Case 3 with thresholding applied reverses this trend and results in the lowest number of revisions overall where almost all revisions incurred minor additional ground delay. Yet, the delay impact was highest under this configuration, implying a potential trade-off between performance and stability, when viewing stability at the flight-impact level.

Finally, we evaluate the TMI strategies generated for this day, where, given the above analyses, we focus on Cases 1 and 3 without thresholding and Case 3 with thresholding to assess the strategies developed. The results are shown in Figures 11-13, respectively, where we note that each resource is distinguished by color and displays the associated rate for a

Table 4. Average GDP Inflection Points

Case	Threshold	# Inflections	# GDP Days	Avg.
Case 1	NT	45	31	1.45
	TH	37	34	1.09
Case 2	NT	52	34	1.53
	TH	49	35	1.40
Case 3	NT	16	35	.46
	TH	18	35	0.50

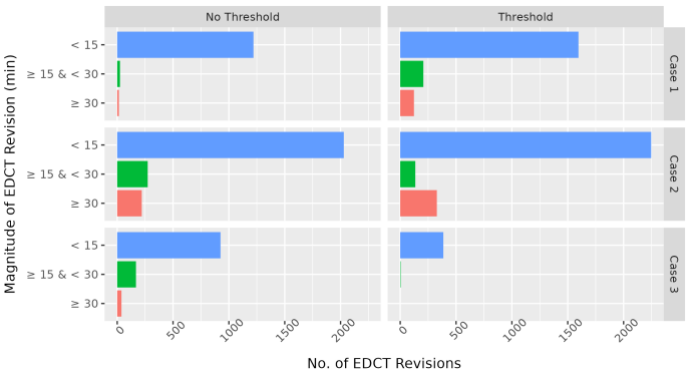

Figure 10. Comparison of June 8th EDCT Revision Delay

GDP (airport, shown in blue) and metering actions at each fix. Each entry has three features: 1) a triangle denoting the time the agent recommended the action, 2) a dashed line signaling the offset between the decision time and when the rate restriction was applied at the corresponding resource, and 3) a solid line indicating the duration of the restriction.

Figure 11 corresponds to the baseline (Case 1 without thresholding), where we note that the GDP strategy varies considerably, especially between the hours of 8-11 local time. During this period, a recommendation is made each hour, where at 9am the first revision decreases the rate, at 10am the second revision significantly increases the rate and at 11am where the third revision decreases the rate to just above the original TMI. This is an undesirable plan from both the perspective of a TM and of a flight-operator and is the behavior we aim to correct with the modifications considered in this paper.

Figure 12 displays the TFM strategy for Case 3 without thresholding, where we see that the agent recommended fewer GDPs with a more consistent rate profile over time. Specifically, there are only 2 inflection points in the GDP rate profile (as compared to three above). Furthermore, the metering at any given fix is mostly set to the same rate throughout the day. Thus, while there is still some rate variability and resulting EDCT revision impacts that are undesirable, this TFM strategy incurred the lowest delay impact on this day.

Figure 13 displays the TFM strategy for Case 3 with thresholding, where we note that two separate GDPs were issued. The first is a short GDP between 10:00 and 12:00. The second GDP decision is made at 12:00 but cannot start until 16:00, due to the restrictions imposed by Case 3, namely a four-hour gap between GDPs. Given the severity in capacity reduction on this day, the decision to wait on a revision, coupled with the restrictions on issuing a second GDP resulted in the worst delay across all cases for this day.

D. Discussion

In our discussions with SMEs, the primary concern for TFM strategy stability centered around when and how GDP decisions were made and revised, as the impact of changes cascade and disrupt the operation in ways that are difficult to quantify. In response, we proposed Case 3 to evaluate the necessity of completely restricting GDP changes, as opposed to limiting the range of revision options. From the results, we see that Case 2 (range restrictions on both GDP and metering) were not only insufficient at improving the performance of the TFM strategy in terms of the stability measures, but these constraints also resulted in poorer performance across all metrics and measures considered. On the other hand, the hard constraints on GDP revisions imposed by Case 3 yielded the best delay impact result for our worst day and generally provided good solutions with greater stability in terms of fewer GDPs and an improved rate profile across the strategy. The tradeoff here, at least on 8 June, was in the number of EDCT revisions issued and points to the potential importance of directly capturing unwanted behavior in the reward function as opposed to the search constraints.

It is important to highlight a limitation of the current model–namely, that the agent can only define a single rate GDP.

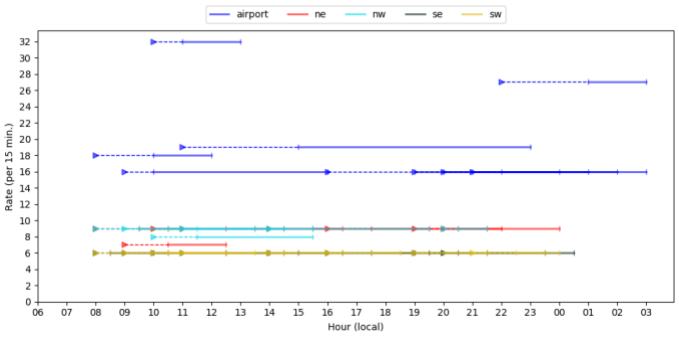

Figure 11. Case 1 TFM strategy without thresholding

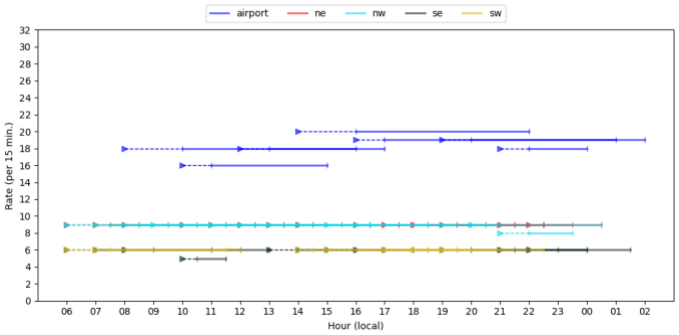

Figure 12. Case 3 TFM strategy without thresholding

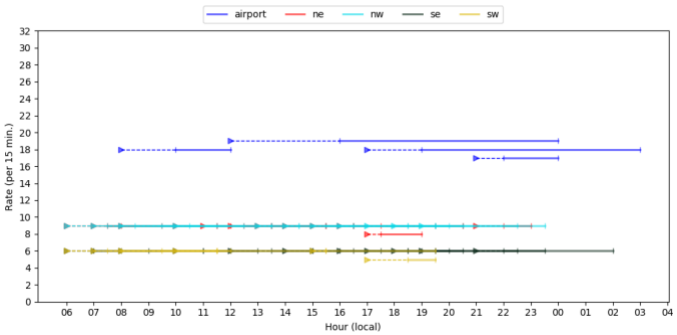

Figure 13. Case 3 TFM strategy with thresholding

However, as we showed in the Case 3 results (without thresholding) limiting the design space can improve the agent's recommendation as there are fewer options to search, especially further into the tree. Thus, care needs to be taken in how a multi-rate GDP is implemented to balance the reality of the operation and the quality of the resulting TFM strategy returned. We will explore different approaches for capturing this complexity adequately in future work.

Furthermore, the threshold on performance improvement demonstrated the potential value, in terms of stability, from waiting, but not on days when a timely response was critical. While future research could explore ways to predict whether the current day falls into one category or another, ultimately, we see this is as a user-set threshold.

Finally, for real-time decision support, the computational time required to identify good decisions needs to be shortened. The results presented in this paper employ software models designed for research purposes, not production and therefore improvements in both the algorithmic implementation and computing infrastructure could enhance the performance with respect to search duration.

VI. Conclusions

This paper demonstrated that considerations that impact the subjective goodness of an agent-generated strategy can be captured using constraints on the search space. While constraint set specifics (e.g., the rate change allowance range) may require further tuning, developing such a mechanism opens possibilities for other interventions, potentially even permitting future TMs to direct the agent in its search and thereby improving calibrated trust in the automation itself. Furthermore, the utility of setting a threshold on performance, while varied, points to an important consideration for how the automation can help guide a TM as to when and to what degree such limitations should be applied. Continuing work will focus on developing and fine-tuning such mechanisms in support of a future AI-based decision support capability for TFM.

Notice

References

[1] Richetta, O. and Odoni, A., "Dynamic Solution to the Ground Holding Problem in Air Traffic Control" *Transportation Research Part A*, Vol. 28, No. 3, May 1994, pp. 167-185. https://doi.org/10.1016/0965-8564(94)90015-9

[2] Ball, M., Hoffman R., Odoni, A. and Rifkin, R., "A Stochastic Integer Program with Dual Network Structure and its Application to the Ground Holding Problem", *Operations Research*, Vol 51., 167-171, 2003. https://doi.org/10.1287/opre.51.1.167.12795

[3] Mukherjee, A., "Dynamic Stochastic Optimization Models for Air Traffic Flow Management with En Route and Airport Capacity Constraints." *The 6th U.S.A./Europe Air Traffic Management Research and Development Seminar*, Baltimore, MD, June 2005.

[4] Liu, P-C. B., Hansen, M., and Mukherjee, A., "Scenario-based Air Traffic Flow Management: From Theory to Practice." *Transportation Research Part B: Methodological*, Vol. 42, No. 7, 2008, pg. 685-702. https://doi.org/10.1016/j.trb.2008.01.002

[5] Hoffman, R., Krozel, J., Davidson, G., and Kierstead, D., "Probabilistic Scenario-Based Event Planning for Traffic Flow Management" *AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, August 2007. https://doi.org/10.2514/6.2007-6361

[6] Steiner, M., Bateman, R., Megenhardt, D., Liu, Y., Xu, M., Pocernich, M., and Krozel, J., "Translation of Ensemble Weather Forecasts into Probabilistic Air Traffic Capacity Impact," Air Traffic Control Quarterly, Vol. 18, No 3, pp. 229-254, 2010. doi:10.2514/atcq.18.3.229.

[7] Taylor, C., Masek, T., Wanke, C., Roy, S., "Designing Traffic Flow Management Strategies Under Uncertainty", *The 11th U.S.A./Europe Air Traffic Management Research and Development Seminar*, Lisbon, Portugal, 2015.

[8] S-L. Tien, H. Tang, D. Kirk, E. Vargo, and S. Liu, "Deep Reinforcement Learning Applied to Airport Surface Movement Planning", *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 8-12 Sept. 2019, San Diego, CA, doi: 10.1109/DASC43569.2019.9081720.

[9] M. Brittain and P. Wei, "Autonomous Air Traffic Controller: A Deep Multi-Agent Reinforcement Learning Approach," *arXiv preprint* arXiv:0902.0885, 2019.

[10] Pham, D-T, Tran N. P., Alam, S., Duong, V., and Delahaye, D., "A Machine Learning Approach for Conflict Resolution in Dense Traffic Scnearios with Uncertainties", 13th USA/Europe Air Traffic Management R&D Seminar, June 2019, Vienna Austria, Paper #18.

[11] Tran, P. N., Pham, D-T., Goh, S. K., Alam, S., Duong, V., "An Interactive Conflict Solver for Learning Air Traffic Conflict Resolutions", AIAA Journal of Aerospace Information Systems, Vol., 17, No. 6, June 2020. DOI. 10.2514/1.I010807

[12] Xie, Y., Pongsakornsathien, N., Gardi, A. and Sabatini, R., 2021. Explanation of machine-learning solutions in air-traffic management. Aerospace, 8(8), p.224.

[13] J. Jones, Z. Ellenbogen, and Y. Glina, "Recommending Strategic Air Traffic Management Initiatives in Convective Weather", 14th USA/Europe Air Traffic Management R&D Seminar, September 2021, Virtual.

[14] C. Taylor, E. Vargo, E. Bromberg, and E. Carson, "Reinforcement Learning for Traffic Flow Management Decision Support", 14th USA/Europe Air Traffic Management R&D Seminar, September 2021, Virtual.

[15] C. Taylor, E. Vargo, E. Bromberg, and T. Manderfield, "Designing Traffic Management Strategies using Reinforcement Learning Techniques", AIAA Aviation Forum, June 2022, Chicago, IL.

[16] https://www.fly.faa.gov/Information/east/ztl/atl/frames.htm, Accessed 7 April 2021.

[17] https://www.faasafety.gov/gslac/alc/libview_normal.aspx?id=9091. Accessed 7 April 2021.

[18] Tien, S-L, Taylor, C., Wanke, C., "Identifying Representative Weather Scenarios for Flow Contingency Management" *AIAA Aviation Forum*, 12-14 Augut 2013, Los Angeles, CA, doi: 10.2514/6.2013-4216.

[19] Bright, D. and Nutter, P., "On the Challenges of Identifying the "Best" Ensemble Member in Operational Forecasting," *84th AMS Annual Meeting*, Seattle, WA, January 2004.

[20] FSM 9.0 Algorithm Specification. Version 1.0. December 17, 2010. Prepared for FAA by CSC. "FSM Algorithm Specifications Ver 1.0 2010-12-17.pdf".

[21] Wan, Y., Taylor, C., Roy, S., Wanke, C., and Zhou, Y., "Dynamic Queuing Network Model for Flow Contingency Management," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1380-1392, Sept. 2013, doi: 10.1109/TITS.2013.2260745.

[22] S. Ontanon, "Combinatorial Multi-armed Bandits for Real-Time Strategy Games", *Journal of Artifiical Intelligence Research*, March 2017, Vol. 58, doi: 10.1613/jair.5398.

Christine P. Taylor is a Principal Artificial Intelligence Engineer at MITRE specializing in decision support system development for traffic flow management applications. She holds a B.S. from Cornell University, and M.S. and Ph.D. degrees in aeronautical engineering from the Massachusetts Institute of Technology.

Erik Vargo is a Principal Artificial Intelligence Engineer at The MITRE Corporation in McLean, Virginia. His research interests include the application of machine learning and probabilistic models to problems in aviation and beyond. He received his PhD in Systems Engineering from the University of Virginia in 2013.

Tyler Manderfield is an Intermediate Data Scientist at The MITRE Corporation specializing in predictive machine learning and data visualization. He received an M.S. in Data Science and a B.A. in Applied Statistics from the University of Virginia.

Simon Heitin is a Lead Data Scientist at The MITRE Corporation in McLean, Virginia. His research applies causal inference methods and data visualization to aviation system performance studies. He holds a B.A. in Mathematics from Yale University and an M.S. in Systems Engineering from George Washington University.