Optimal Control of Airport Pushbacks in the Presence of Uncertainties

Patrick McFarlane[1] and Hamsa Balakrishnan[2]

***Abstract*— This paper analyzes the effect of a dynamic programming algorithm that controls the departure pushback rate at congested airports, with an emphasis on the uncertainty of the underlying processes. The state of the airport at any time includes the number of departures taxiing to the runway, and the number of departures queued at the runway. The state of the airport surface at the start of a time-window is used to calculate the probability distribution of the state at the end of that time-window, accounting for uncertainties in the system. A cost function that penalizes both excessively long as well as empty runway queues is used to determine the optimal pushback rate for that time-window, using dynamic programming. Since the level of uncertainty in the system increases with the length of time-window, the performance of the dynamic programming policy is evaluated, for different lengths of time-window and planning time horizon. Uncertainty in both arrival and departure demand parameters are evaluated in simulation. Case studies of LaGuardia International Airport (LGA) shows that the dynamic programming algorithm can potentially reduce the departure taxi-out time by over 175,000 minutes over a 2-month period, even in the presence of arrival and departure demand uncertainties, and a planning horizon of 45 minutes.**

***Index Terms*— Air traffic control, airport surface management, dynamic programming**

I. INTRODUCTION

Airlines operated 7.7 million departures from major airports in the United States and consumed 10.3 billion gallons of fuel in 2014 [1], [3]. Such high levels of traffic can lead to congestion, resulting in delays. The average taxi-out delay across the entire U.S. is greater than 5 min, with LaGuardia (LGA) Airport in New York leading the nation with an average taxi-out delay of more than 12 min [2]. Congestion results in long queues of aircraft at the departure runways, resulting in additional costs in terms of fuel burn and emissions.

Delays due to long departure queues can be reduced or redistributed through the control of departing aircraft. However, careful consideration must be given to the uncertainties present in the operating environment. The effectiveness of control policies can depend on these uncertainties, as well as the accuracy of the input data. This paper develops and applies a congestion control algorithm to aircraft pushbacks, and evaluates its performance in the presence of uncertainties.

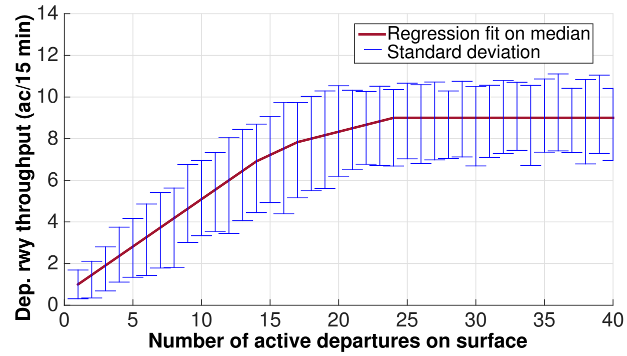

Fig. 1. Departure throughput versus departure aircraft taxiing at LaGuardia Airport for runway configuration 31|4, under Visual Meteorological Conditions (VMC).

Departing aircraft push back from their gates, taxi to their assigned runway, and wait in the departure queue until takeoff. Surface management algorithms control the aircraft operating at an airport so as to reduce congestion while maintaining the throughput of the runways. Fig. 1 shows the departure runway throughput as a function of the level of departure traffic on the surface, using 2013 data at LGA. At low levels of surface traffic, increasing aircraft pushbacks (i.e., the level of surface traffic) results in a corresponding increase in departure throughput. However, after a certain traffic level, an increase in traffic no longer results in an increase in departure throughput, and merely leads to congestion. Several control algorithms have been developed to mitigate congestion by controlling the rate at which departing aircraft push back from their gates. These algorithms are called Pushback Rate Control (PRC) policies. They work within the existing airline schedules by holding certain flights during times of congestion.

Dynamic programming can be used to determine PRC policies that control the departure pushback rate by minimizing a cost function that penalizes long runway queues and low runway utilization [13]. For each time-window (say, 15 min) and given the current state of the system, a queuing model is used to predict the state of the system at the end of that time-window. A prediction of the departure throughput of the airport for that time-window (which in turn depends on the arrival rate of the airport) is required in order to predict the state of the airport. The dynamic program then determines an optimal pushback rate for the duration of the time-

*This work was supported through MIT Lincoln Laboratory by the FAA's Office of Environment and Energy under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

[1]Patrick McFarlane is a graduate research assistant in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology pmcfar@mit.edu

[2]Hamsa Balakrishnan is an associate professor in the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology hamsa@mit.edu

window. The dynamic programming policy has the benefit that it accommodates probabilistic predictions of throughput. However, the uncertainty associated with the throughput prediction increases if either the time-window increases (say, to 30 min instead of 15 min), or if the time horizon increases (for example, planning for the time-windows that begin 15- or 30-min later instead of just at the current time). On the demand-side, arrival rate predictions may not be accurate, which will in turn increase the uncertainty associated with the predicted departure throughput. Similarly, departure demand may vary, and aircraft may not push back at the times recommended by the pushback rate. The impact of all these uncertainties are evaluated using simulations of operations at LGA.

II. RELATED RESEARCH

Airport surface management research has shown that pushback control policies lead to reductions in fuel burn and emissions, minimizing the impact of surface operations [5]–[8]. Simaiakis and Balakrishnan developed and simulated a threshold policy, N-control, for Boston Logan International (BOS) airport and reported important metrics to describe the effects of airport surface management [5]. They also calculated those metrics for several major airports and analyzed the resultant reductions in emissions and fuel burn [6]. Ravizza et al. demonstrated the relationship between airport surface movement and fuel burn [7]. Khadilkar examined the control of both departures and arrivals on an individual aircraft basis [8].

Simaiakis developed methods for the estimation of airport capacity and unimpeded taxi-out times, and a dynamic programming algorithm for BOS [9], [10]. These techniques form the basis of the research presented in this paper, to develop models for LaGuardia (LGA) airport, and to study the impact of uncertainties and policy parameter selection on the performance of PRC policies. Simaiakis also noted that PRC policies are flow-based approaches to airport surface management, which means that these policies use virtual queues by holding aircraft at their gates. The use of virtual queues were initially suggested [11] and proposed in separate studies [12]. Feron et al. [11] give a detailed overview of the conceptual departure control process, culminating in the idea of using virtual queues to mitigate congestion. Burgain et al. [12] use virtual queues to minimize a cost function related to passenger wait time.

Departure metering policies have also been determined using dynamic programming [13]. This method accounts for the underlying uncertainty of the airport capacity by modeling the state of the airport surface as a semi-Markov process. The optimal pushback rate is calculated based on a cost of queuing function and the probability of the airport surface being in a given state. However, this analysis did not examine the effect of the choice of policy parameters, such as the time-window over which a pushback rate remains valid, and the planning horizon. This paper fills in this gap. Other relevant work on dynamic programming applied to airport operations has only focused on the optimization of aircraft scheduling. Rathinam et al. [14] proposed a dynamic programming approach for the departure schedule that finds the optimal pushback schedule for a given number of departing aircraft. Chandran and Balakrishnan [15] also used a dynamic programming algorithm for the departure runway schedule, but they only accounted for the uncertainty and random deviations inherent in the runway process. Dell'Olmo and Lulli considered the allocation of arrival and departure capacities using dynamic programming, and the tradeoffs between them [16].

This paper describes the development of a dynamic programming algorithm for pushback rate control at LGA, and examines the impact of the length of the time-window (the length of time for which a pushback rate remains valid) and time horizon (the number of time-windows for which a pushback rate is calculated) on policy performance. The analysis also considers the uncertainties pertaining to the departure schedules and arrival rates. The results, based on a 2-month simulation of LGA operations, show the considerable benefits airlines would receive under the dynamic programming-based policy, even in the presence of uncertainty.

III. ALGORITHM DEVELOPMENT

The dynamic programming algorithm models the state of the airport surface as a Markov process with the state described by the number of aircraft taxiing to the runway and the number of aircraft queuing at the runway. By modeling the runway service times as an Erlang distribution [10] with the shape and rate $(k, k\mu)$, the transition probabilities over a time-window are found by numerically integrating the Chapman-Kolmogorov equations, which are described below. The runway service time is the time between successive takeoffs on a runway, meaning that a service time is the time it takes the aircraft at the head of the queue to leave the airport surface. Dynamic programming then uses value iteration to find the optimal pushback policy in terms of the costs of queuing and runway utilization.

To account for different runway service times due to variability in the number of arrivals or weather, the machine learning technique of regression trees can calculate departure throughput under many different conditions. The regression trees only describe the throughput for periods of congestion, corresponding to the flat portion of Fig. 1. Using empirical data, the regression trees calculate the predicted departure throughput based on the arrival rate and route availability during times of congestion. The Route Availability Planning Tool (RAPT) is a tool that uses predictive algorithms to estimate the location and severity of weather in the area surrounding an airport [17]. RAPT is incorporated in the regression trees by averaging values for a 30 min period over all departure routes. Fig. 2 shows an example of a regression tree for LGA.

A regression tree to predict departure runway throughput is found for each runway configuration and weather condition, or segment. For each leaf of the regression trees, the shape and rate, unknown parameters of an Erlang distribution, are

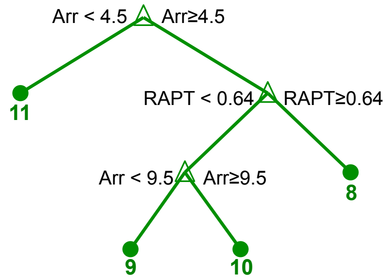

Fig. 2. Regression tree with predicted departure throughput at the leaves for LGA for segment (31|4; VMC) with a 15 min time-window.

found by fitting empirical runway service times using the method of moments. For an airport, a solution is found for each shape of the Erlang distribution of service times.

At the beginning of the time-window, the state (r, q) is observed, where r is the number of aircraft taxiing to the runway and q is the stages-of-work to be completed at the runway. The stages-of-work are the product of the number of aircraft queuing and the shape of the Erlang distribution. With the state (R_0, Q_0) at the beginning of the time-window and the runway capacity C, the following Chapman-Kolmogorov equations are then solved for the entire time-window of length Δ to get the probability $P_{r,q}(\Delta)$ of the airport surface being in a given state at the end of the time-window [10]:

$$\frac{dP_{0,0}}{dt} = k\mu P_{0,1}, \tag{1}$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} - k\mu P_{0,q},\ 1 \leq q < k, \tag{2}$$

$$\frac{dP_{0,q}}{dt} = k\mu P_{0,q+1} + \frac{1}{\Delta - t} P_{1,q-k} - k\mu P_{0,q},\ k \leq q < kC, \tag{3}$$

$$\frac{dP_{0,kC}}{dt} = \frac{1}{\Delta - t} P_{1,k(C-1)} - k\mu P_{0,kC}, \tag{4}$$

$$\frac{dP_{r,0}}{dt} = k\mu P_{r,1} - \frac{r}{\Delta - t} P_{r,0}, \tag{5}$$

$$\frac{dP_{r,q}}{dt} = k\mu P_{r,q+1} - k\mu P_{r,q} - \frac{r}{\Delta - t} P_{r,q},\ 1 \leq q < k, \tag{6}$$

$$\begin{aligned}\frac{dP_{r,q}}{dt} &= k\mu P_{r,q+1} + \frac{r+1}{\Delta - t} P_{r+1,q-k} - k\mu P_{r,q}, \\ &\quad - \frac{r}{\Delta - t} P_{r,q},\ k \leq q \leq k(C-1),\end{aligned} \tag{7}$$

$$\begin{aligned}\frac{dP_{r,q}}{dt} &= k\mu P_{r,q+1} + \frac{r+1}{\Delta - t} P_{r+1,q-k} - k\mu P_{r,q}, \\ &\quad k(C-1) < q < kC,\end{aligned} \tag{8}$$

$$\frac{dP_{r,kC}}{dt} = \frac{r+1}{\Delta - t} P_{r+1,k(C-1)} - k\mu P_{r,kC}, \tag{9}$$

$$\frac{dP_{R_0,0}}{dt} = k\mu P_{R_0,1} - \frac{R_0}{\Delta - t} P_{R_0,0}, \tag{10}$$

$$\begin{aligned}\frac{dP_{R_0,q}}{dt} &= k\mu P_{R_0,q+1} - (\frac{R_0}{\Delta - t} + k\mu) P_{R_0,q}, \\ &\quad 1 \leq q \leq k(C-1),\end{aligned} \tag{11}$$

$$\frac{dP_{R_0,q}}{dt} = k\mu P_{R_0,q+1} - k\mu P_{R_0,q},\ k(C-1) < q < kC, \tag{12}$$

$$\frac{dP_{R_0,kC}}{dt} = -k\mu P_{R_0,kC}. \tag{13}$$

With these transition probabilities, the expected cost of releasing aircraft with pushback rate λ can be found, with the assumption that aircraft traveling to the runway queue at the beginning of one time-window reach the queue by the start of the next time-window. The minimum cost $J^*(r, q)$ at each state is given by the Bellman equation for the infinite horizon problem with discount factor α:

$$J^*(r,q) = \min_{\lambda \in \Lambda} \{\bar{c}(r,q) + \alpha \sum_{j=0}^{kC} P_{r,q} J^*(\lambda, j)\}, \tag{14}$$

where $\bar{c}$ is an average cost of a state over a time period and Λ is the set of all possible pushback rates. Here, $P_{r,q}$ is the probability, starting at state (r, q) at the beginning of the time period, of being in a state (λ, j) at time Δ. This equation is solved by value iteration. The pushback rate for the time-window is given by the λ that minimizes the cost function.

The cost function must penalize both non-utilization of the runway and excessively long queues. Non-utilization of the runway is assigned a constant cost H, while for $q > 0$, the cost is a non-decreasing function of q [10]. Therefore

$$c(q) = \begin{cases} H & \text{if } q = 0, \\ (\frac{q-k}{k})^2 & \text{if } q > 0. \end{cases} \tag{15}$$

H is chosen to reflect the true cost of not maintaining runway utilization. $c(q)$ is only a function of queue length and service time shape. Because dynamic programming accounts for all possibilities for the evolution of the airport state, the cost function must be combined with the probability that the runway queue is of a certain length. To add the time component, the vector of these probabilities is

$$\mathbf{p}_q(R_0, Q_0, t) = [\sum_{r=0}^{R_0} P_{r,0}(t), \sum_{r=0}^{R_0} P_{r,1}(t), ..., \sum_{r=0}^{R_0} P_{r,kC}(t)]. \tag{16}$$

In words, the above equation states that, given that the state of the airport was (R_0, Q_0) at the beginning of the time-window, these are the probabilities that the runway queue consists of q stages-of-work at time t. Now, with the probability of runway queue length as a function of time, the product of these probabilities and the cost function can be summed over an entire time-window Δ to find the expected cost of each state:

$$\bar{c}(R_0, Q_0) = \sum_{i=0}^{10\Delta - 1} \frac{1}{10} \mathbf{p}_q(R_0, Q_0, i/10) \cdot c(q). \tag{17}$$

Because (16) is sampled 10 times a minute, the summation in (17) reflects this sampling. With the expected cost over a time-window, (14) is solved to find the optimal pushback rate. The solution over all states can be seen in Fig. 3 for an Erlang distribution of service times with shape $k = 2$ at LGA with a maximum pushback rate of 15 aircraft per 15 min.

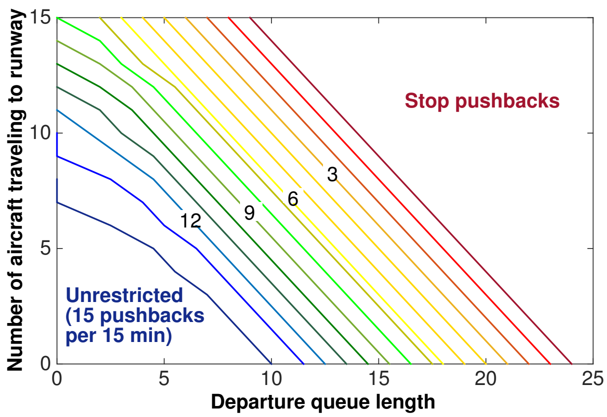

Fig. 3. Parametric solution of pushback rate with observed values of aircraft taxiing and departure queue length for Δ = 15 min. Each line is the optimal pushback rate for the given state, increasing from 0 to 15 from right to left.

Fig. 3 provides an intuitive understanding of the solution determined by the dynamic programming algorithm. For a low departure queue Q_0 and taxiing departures R_0, the airport surface is relatively empty and the dynamic programming recommends a maximum pushback rate. For high Q_0 and low R_0, the pushback rate decreases slower with increasing Q_0 when compared to increasing R_0 for high R_0 and low Q_0. This makes sense because, with high Q_0 and low R_0, the queue will likely diminish by the time the recommended pushback rate reaches the runway queue. If R_0 is high and Q_0 is low, the aircraft taxiing will replenish the queue before the recommended pushback rate reaches the runway. Therefore, the pushback rate does not need to be as high. This is reflected in both the slope and nonlinear characteristics of each parametric solution in Fig. 3.

IV. CHOICE OF ALGORITHM PARAMETERS

The dynamic programming policy generates a pushback rate that is valid for a given time-window of duration Δ. A pushback rate can be calculated for several time-windows (each of length Δ) into the future by changing the time horizon of the policy. In other words, the optimal pushback rate remains constant over a time-window, while the time-horizon determines how far in advance the pushback rate is determined. Changing the time-window or time horizon allows an airport or airline to tailor the PRC policy to specific needs and requirements, but can also change the performance of the policy.

A. *Duration of time-window*

The duration of the time-window, Δ, refers to the length of time over which a given pushback rate is maintained. The pushback rate is updated periodically by repeating the algorithm described in Section III. Historically, the value of Δ has been set to 15 min [13], but varying it has its advantages and disadvantages.

Because the dynamic program calculates the optimal pushback rate at the beginning of the time-window based on the current state of the airport surface, a longer time-window means that the pushback rate is valid for a longer period of time. As a result, the rate cannot adapt dynamically to changes in throughput or demand. In addition, the uncertainty associated with the probability distribution of departure throughput also increases as Δ increases, and the policies become less accurate, resulting in a reduction in benefits.

The duration of time-windows impacts the workload of air traffic controllers. For shorter time-windows, the controller needs to update the pushback rate more frequently. This also requires the gathering of input data, and the recalculation of the pushback rate. This workload decreases with the increase in the length of time-window because the pushback rates remain valid for a longer length of time. However, as explained above, this decrease in workload could potentially come at the cost of decreased benefits.

B. *Length of time horizon*

Although the pushback rate has previously only been calculated at the start of each time-window, it may be preferable to calculate these rates ahead of time. Suppose the length of time-window is set to 15 min. Then, a time horizon of 45 min would imply that at any time, the pushback rate would be calculated for the time-window that begins 45 min later, that is, the window that extends between 45 min and 1 hour from the current time. These rates would be based on a *prediction* of the airport state 45 min later, which would be more inaccurate than an observation of the current state. As a result, the accuracy of the suggested pushback rate will also decrease, resulting in a decrease in benefits. However, longer time horizons provide estimated pushback rates and pushback times to airlines well in advance, and enable better operations planning on their part.

V. SIMULATION RESULTS

A. *Input data*

The data required for the simulation must be extracted from multiple sources. The ASPM dataset provides flight specific metrics such as pushback time, wheels off time, and wheels on time [1]. Gate and terminal assignments allow for the calculation of unimpeded taxi-out times, and allow the policy to monitor gate conflicts. The last dataset contains the weather data, RAPT, described previously. The simulations consider July and August 2013.

Each simulation contains both a baseline case and a metering case. The baseline case simulates the airport operations by releasing departures from their gates on a First-Come-First-Served (FCFS) basis based on scheduled departure times. The metering case simulates the airport operations using the dynamic programming policy. The benefits of the policy include the taxi-out time reduction, which is the difference between the taxi-out times in the baseline case and metering case. Taxi-out time reduction contrasts with gate holding time, which is the length of time an aircraft is held at a gate beyond the scheduled departure time due to the dynamic programming policy. Gate holding time is not strictly a cost because aircraft still belong to the virtual queue with engines off. However, occupying the gate

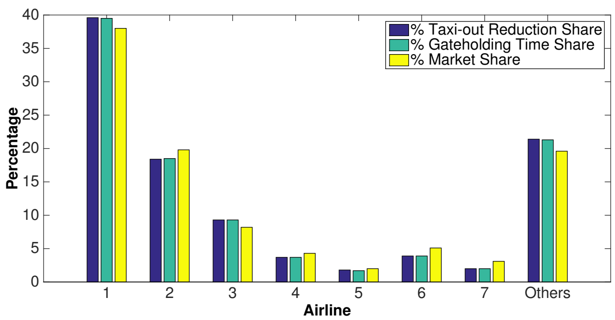

Fig. 4. Percentage taxi-out time reduction, gateholding time and market share by airline for LGA simulations of the dynamic programming policy for July-August 2013. These results are for a 15-minute time-window and zero time horizon.

for longer periods of time causes more gate conflicts, and extended gateholding times after boarding can inconvenience passengers.

Considering the factors discussed in Section IV, the simulated durations of time-windows include the usual 15 min, as well as longer time-windows of 30 min and 60 min. To mirror the time-window analysis, time horizons of 0 (the immediate time-window), 15 min (one time-window in advance) and 45 min (3 time-windows in advance) are considered. As mentioned earlier, a time-horizon of 45 min requires determining a pushback rate for the 15-min time-window that extends between 45 min and 60 min later.

B. Fairness of dynamic programming policy

Fig. 4 shows the percentage of taxi-out time reduction, gateholding time, and market share (in terms of operations) corresponding to each airline, for simulations of the 15-min time-window (and zero time horizon).

Examining Fig. 4 reveals some interesting features of the dynamic programming policy. For each airline, all of the percentages in Fig. 4 are roughly equal. The proportionality between taxi-out time reduction and market share illustrates that the dynamic programming policy is fair across airlines. The benefits that an airline can expect are commensurate with its share of operations. The relationship between taxi-out time reduction and gateholding time reveals the effectiveness of the dynamic programming policy, as airlines do not need to invest a disproportionate amount of gateholding time to realize the benefits of taxi-out time reduction. While Fig. 4 considers the 15-minute time-window, the same fairness properties persist for 30-min and 60-min time-windows.

C. Duration of time-windows

Fig. 5 shows the total taxi-out time reduction for each airline, for simulations of different durations of time-window. It illustrates that the taxi-out time reduction benefits decrease for increasing time-window lengths. Relative to the 15-min time-window length results, a 30-min time-window has 63% of the taxi-out time reduction and a 60-minute time-window has only 43% of the taxi-out time reduction. These results are consistent with the rationale that as the time-window

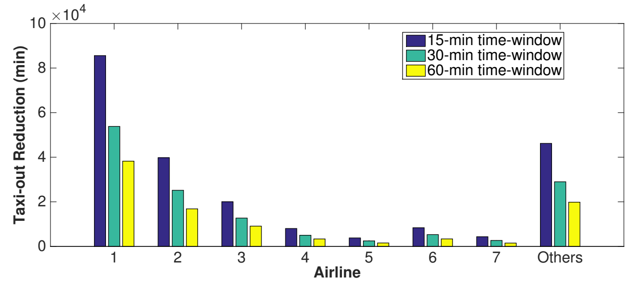

Fig. 5. Total taxi-out time reduction by airline for different values of time-window duration, assuming a zero planning time horizon. Based on simulations of LGA operations in July-August 2013.

becomes longer, the accuracy of the dynamic programming policy decreases, along with its benefits.

D. Length of time horizon

In investigating the impacts of changing the time horizon, the time-window is set to 15-min. Fig. 4 corresponds to the case with no planning time horizon. While details are not discussed here for reasons of brevity, policy fairness and proportionality of benefits and gateholding times are found to be maintained for different values of time horizon.

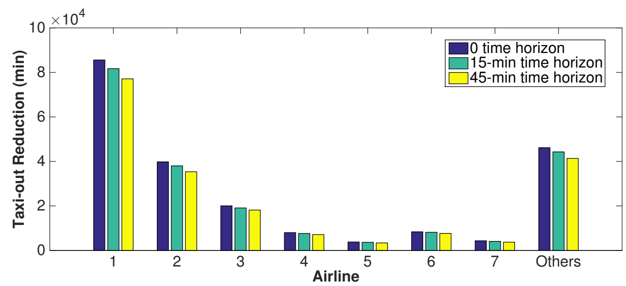

Fig. 6. Total taxi-out reduction by airline for different values of planning time horizon, assuming a 15-min time-window. Based on simulations of LGA operations in July-August 2013.

Fig. 6 shows the total taxi-out time reduction for each airline, for different values of planning time horizon. The deterioration in taxi-out time reduction for increasing time horizon length (Fig. 6) is as not drastic as the decrease with increasing time-window (Fig. 5).

For longer time-windows, the dynamic programming algorithm finds transition probabilities further into the future, and uncertainty in the evolution of the state of the airport surface results in a less accurate departure pushback rate. Secondly, the rate can only adapt at the end of a time-window. For longer time horizons, the initial pushback rate for the first time-window is found as usual. For the following time-windows in the time horizon, the pushback rate for the previous time-window becomes the number of aircraft traveling to the runway in the next time-window. The runway queue is updated as follows:

$$\frac{Q_0^*}{k} = \frac{Q_0}{k} - T_p + R_0, \tag{18}$$

where $\frac{Q_0^*}{k}$ is the new runway queue length, $\frac{Q_0}{k}$ is the runway queue length in the previous time-window, and T_p is the predicted runway throughput in the previous time-window. While this is approximate, even under a longer time horizon, the state of the airport surface is updated throughout a time horizon (at increments of Δ). By contrast, a time-window increase relies on the original state of the airport surface, and a single pushback rate for the entire time-window. This difference results in greater robustness of performance under longer time horizons than under longer time-windows.

E. Uncertainties in arrival and departure demand

1) Departure demand uncertainty and conformance to target pushback times: There are two sources of uncertainty in departure demand: First, that the demand does not materialize as expected (in other words, aircraft call ready for pushback at times that are different from their scheduled or reported earliest pushback times[1]), and second, that departures do not pushback at the target times that are assigned to them. The impacts of either of these cases can be simulated by considering perturbations of pushback times, since they both translate to aircraft being ready for pushback at a time other than their EOBT.

The perturbation to the departure (pushback) time of a flight is drawn from a normal distribution with standard deviation of 3.5 min. The assumption of a standard deviation of 3.5 min ensures that two standard deviations of the perturbation encompass an approximately 15-min span of time.

For each flight, a perturbation time is randomly drawn from the probability distribution described above and added to the original departure time. The new departure time of a flight t^* is given by

$$t^* = t + t_p, \tag{19}$$

where t is the scheduled departure time of the flight and t_p is the perturbation. The scheduled departure time for each flight in a day is updated using (19). The simulation of airport operations with the the dynamic programming policy then proceeds as usual.

The results of the simulation with the original schedule can be compared to the results of the simulation with the perturbed schedule. However, because the perturbed schedule is subject to random sampling, the results of the simulation with the perturbed schedule are also subject to the random sampling. As such, the variable departure schedule uses a Monte Carlo method to get a better sense of the results from perturbing the schedule. The Monte Carlo simulation reruns each day over the 2 months with a PRC policy 50 times, each time with a different perturbed schedule. The results are the average values over the 50 trials.

2) Arrival rate uncertainty: Similar to the departure schedule perturbations, an arrival rate perturbation for a 15 min time-window is drawn from an arrival rate distribution with mean 0 and standard deviation of 1, which is similar to

[1]Also known as Earliest Off-Block Times, or EOBT

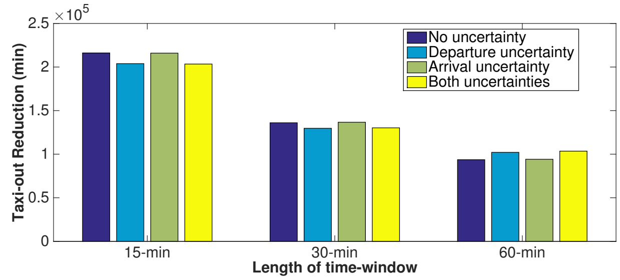

Fig. 7. Total taxi-out time reduction with and without uncertainties for different values of time-window duration, assuming a zero planning time horizon. Based on simulations of LGA operations in July-August 2013.

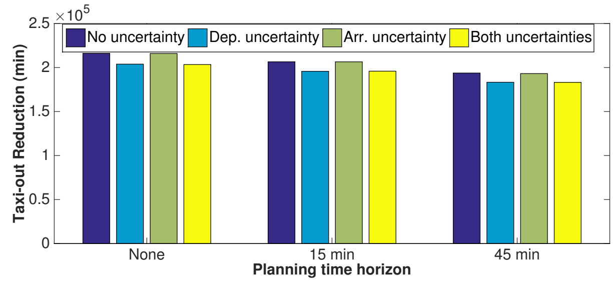

Fig. 8. Total taxi-out reduction with and without uncertainties for different values of planning horizon, assuming a 15-min time-window. Based on simulations of LGA operations in July-August 2013.

what is seen in practice. For the 30-min and 60-min time-windows, the standard deviations of the arrival rate perturbations are assumed to be 2 and 4, respectively. Because arrival rates reflect the number of aircraft expected to land in a certain time-window, the arrival rate perturbation is rounded to the nearest whole number. The new arrival rate a^* then becomes

$$a^* = a + a_p, \tag{20}$$

where a is the original arrival rate and a_p is the arrival rate perturbation. (20) updates all of the original arrival rates for each time-window. These perturbations result in new values of the predicted departure throughput in each time-window, since the regression trees (e.g., Fig. 2) depend on the arrival rate. 50-Trial Monte Carlo simulations with different arrival schedules are conducted, and the results are averaged.

3) Impacts of uncertainty: Figs. 7 and 8 show the results of the dynamic programming policy with both arrival and departure uncertainties, with different choices of time-window and planning horizon.

Figs. 7 and 8 show similar trends for variations of time-window and planning horizon – the benefits are more robust to changes in the planning horizon than the time-window. In both cases, the departure uncertainty has a greater impact on benefits than the arrival rate uncertainty. In all but one case, the departure uncertainty decreases the benefits, while arrival rate uncertainty has little to no effect on taxi-out time reduction. For the 60-minute time-window, the departure uncertainty increases the policy benefits slightly, likely due to the spreading out of scheduled departure times (e.g., (19)). Simulations with both sources of uncertainty show that the dynamic programming algorithm handles uncertainty very well, and yields benefits that are nearly equal to the case with no uncertainty. More importantly, the approach has the potential to yield very significant benefits, nearly 175,000 min over the two-month period, even in the presence of arrival and departure uncertainties. These benefits could be achieved while giving airlines target pushback times at least 45 min in advance of operations.

VI. CONCLUSIONS

This paper explored the impacts of time-window duration, advance planning horizon and uncertainties on the performance of a dynamic programming based airport congestion control strategy. The dynamic programming approach considered the state of the airport surface at the current time, and calculated an optimal pushback rate for the duration of a time-window, accounting for operational uncertainties. The control strategy is evaluated using simulations of operations at LGA for a two-month period. The results indicate that the performance of the policy deteriorates significantly as the time-window increases from 15 min to 60 min. They also show that the control policy performs well even if the planning horizon is increased to 45 min, with a time-window of 15 min. Departure demand variability is found to have a much larger impact on the taxi-out time benefits than arrival rate variability. The simulations show that the dynamic programming algorithm can potentially reduce the departure taxi-out time at LGA by over 175,000 min over a 2-month period, even in the presence of arrival and departure demand uncertainties. These benefits can be achieved while providing airlines with target pushback times 45 min in advance, which is promising from the perspective of airline operations planning.

ACKNOWLEDGMENTS

We would like to thank Hector Fornés Martinez (MIT) for his help with processing airport operations data for LGA, and Armen Samurkashian (MIT) for his help in estimating arrival rate uncertainty. We also thank Ngaire Underhill (MIT Lincoln Laboratory) for her help with RAPT data.

REFERENCES

[1] U.S. Department of Transportation, Aviation System Performance Metrics, 2014.

[2] D. Chin and M. Castelletti, Comparison of air traffic management-related operational performance: U.S./Europe, Jun 2014.

[3] Bureau of Transportation Statistics Form 41 Schedule P12A as of 9/16/2015.

[4] Joint Economic Committee. Flight delays cost passengers, airlines, and the U.S. economy billions, May 2008.

[5] I. Simaiakis and H. Balakrishnan, Queuing models of airport departure processes for emissions reduction. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2009.

[6] I. Simaiakis and H. Balakrishnan, Impact of congestion on taxi times, fuel burn, and emissions at major airports. *Transportation Research Record: Journal of the Transportation Research Board*, 2184:22-30, 2010.

[7] S. Ravizza, J. Chen, J. Atkin, E. Burke, and P. Stewart. The trade-off between taxi time and fuel consumption in airport ground movement. *Public Transport*, 5(1-2):25-40, 9 2013.

[8] Analysis and modeling of airport surface operations. Master's thesis, Massachusetts Institute of Technology, 2011.

[9] I. Simaiakis, Modeling and control of airport departure processes for emissions reduction. Master's thesis, Massachusetts Institute of Technology, 2009.

[10] I. Simaiakis. Analysis, Modeling and Control of the Airport Departure Process. PhD thesis, Massachusetts Institute of Technology, 2009.

[11] E. R. Feron, R. J. Hansman, A. R. Odoni, R. B. Cots, B. Delcaire, W. D. Hall, H. R. Idris, A. Muharremoglu, and N. Pujet. The departure planner: A conceptual discussion. Technical report, Massachusetts Institute of Technology, 1997.

[12] P. Burgain, E. Feron, J. Clarke, and A. Darrasse. Collaborative virtual queue: Fair management of congested departure operations benefit analysis. *arXiv Preprint*, (arXiv:0807.0661), 2008.

[13] I. Simaiakis, M. Sandberg, and H. Balakrishnan. Dynamic control of airport departures: Algorithm development and field evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):285-295, 2 2014.

[14] S. Rathinam, Z. Wood, B Sridhar, and Y. Jung. Generalized dynamic programming approach for a departure scheduling problem. *AIAA Guidance, Navigation, and Control Conference*, 6250:1-12, 2009.

[15] B. Chandran and H. Balakrishnan. A dynamic programming algorithm to robust runway scheduling. *American Control Conference*, 6250:1161-1165, 2007.

[16] P. Dell'Olmo and G. Lulli. A dynamic programming approach for the airport capacity allocation problem. *IMA Journal of Management Mathematics*, 14:235-249, 2003.

[17] R. DeLaura, M. Robinson, R. Todd and K. MacKenzie. Evaluation of weather impact models in departure management decision support: Operational performance of the Route Availability Planning Tool (RAPT) prototype. *13th Conference on Aviation, Range, and Aerospace Meteorology*, 2008.