Deep Reinforcement Learning Based Airport Departure Metering

Hasnain Ali, Pham Duc Thinh and Sameer Alam

***Abstract*— Airport taxi delays adversely affect airports and airlines around the world in terms of congestion, operational workload, and environmental emissions. Departure Metering (DM) is a promising approach to contain taxi delays by controlling departure pushback times. The key idea behind DM is to transfer aircraft waiting time from taxiways to gates. State-of-the-art DM methods use model-based control policies that rely on airside departure modeling to obtain simplified analytical equations. Consequently, these models fail to capture non-stationarity existing in the complex airside operations and the policies perform poorly under uncertainties. In this work, we propose model-free and learning-based DM using Deep Reinforcement Learning (DRL) approach to reduce taxi delays while meeting flight schedule constraints. We cast the DM problem in an MDP framework and develop a representative airport simulator to simulate airside operations and evaluate the learnt DM policy. For effective state representation, we introduce features to capture both local and airport-wide congestion levels. Finally, the performance of multiple agents-sharing the same trained policy, is evaluated on different traffic densities. The proposed approach shows a reduction of up to 25% in taxi delays in medium traffic scenarios. Moreover, upon experiencing increased traffic density, taxi time savings achieved by proposed algorithm significantly increase while the average gate holding times do not increase as much. Results demonstrate that DRL can learn an effective DM policy to better manage airside traffic and contain congestion on the taxiways.**

I. INTRODUCTION

Airports have been identified as key choke points in the air transportation system causing major delays [1]. Air traffic growth, in the absence of airport capacity expansions- which are cost intensive and environmentally sensitive, is expected to increase congestion and snowball delay effects at airports. Airport taxi delays, in the US alone, cost approximately $900 million every year due to extra fuel burn [2]. For airlines, in addition to unnecessary fuel burn, this translates to lower operational efficiency. Moreover for passengers, delays lead to poor travel experience and sometimes missed connections [3], [4]. Growing carbon foot-prints of congested airports worldwide have exacerbated environmental concerns as well [5].

Departure metering (DM) offers an opportunity to contain airside congestion and delays by controlling departure pushback time or target start up approval time (TSAT). DM is an airside surface management procedure in which departures are held at gates to reduce airside congestion and are appropriately released before their take-off time. This is done to help departing aircraft reach runway just in time for take-off while preventing large queue formations on the airside. The key idea behind DM is to transfer waiting time from taxiway, where the engine is switched on and therefore burning fuel, to gates where the aircraft is on auxiliary power mode [6]. Due to DM, aircraft waiting time at the taxiway and runway queues can be reduced which in turn may lower fuel consumption, related emissions and improve the airside traffic circulation by having a smoother departure flow. Since smaller departure queues are formed on the airside (refer Fig. 1), ground trajectories of arrivals face fewer obstructions in their intended movement toward gates. This lowers chances of formation of congestion hotspots in the airside network, which in turn reduces Air Traffic Controller's (ATCO) workload [7].

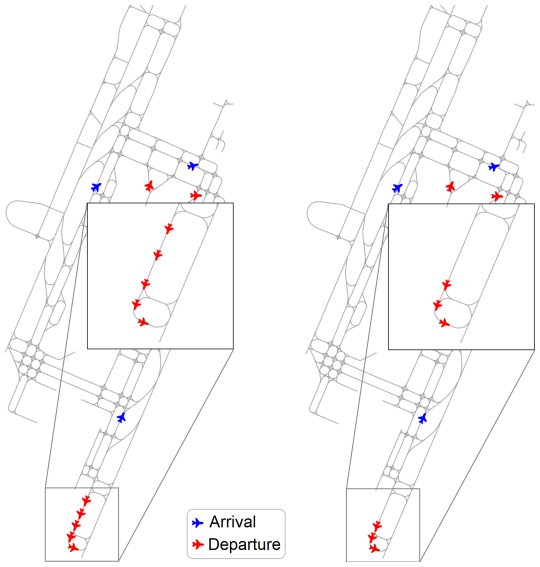

Fig. 1: Demonstration of departure metering potential to mitigate surface congestion at Changi airport (red and blue icons depict departure and arrival aircraft respectively). Left: Longer queue formation before airside surface congestion management; Right: Shorter queue formation after airside surface congestion management.

Recent approaches have employed handcrafted multi-integer linear programming models [8], cell transmission models [9], queuing models [10], statistical models [7] etc. to model airside traffic flow. State of the art DM methods use these models to derive control policies for performing DM. These control policies dictate TSAT of departure aircraft. To obtain analytical equations, such model based DM methods make many simplifying assumptions like assuming that queuing occurs only at the runway and not on the taxiways [11]. As such, these attempts at modelling the airside environment fail to capture non stationarity

Hasnain Ali, Pham Duc Thinh and Sameer Alam are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. {hasnain001, dtpham, sameeralam}@ntu.edu.sg

and uncertainty existing in the complex airside operations. Moreover, control actions, that are derived from DM control policies, are based on model parameters- which are often fixed, fail to handle large airside environment perturbations. Not surprisingly, benefits realised due to these methods have reduced significantly under uncertainty [12], [13].

DM is a sequential decision-making problem where the decision to release a departing aircraft from the gate not only impacts localised congestion around apron but also affects movement of arrivals and queuing of departures near runways at a later point in time. Reinforcement learning (RL) can be potentially used to solve sequential decision-making problems. Notable achievements of RL include playing sequential decision-making games like chess at super human level. To the best of our knowledge, only [14] till date has investigated the potential of RL to devise optimal DM policy. But state space in their MDP formulation consisted of only one feature- the number of aircraft ahead in the runway queue. We believe this makes the state representation shallow and insufficient to effectively capture the evolution of airside traffic. Recently, the combination of deep learning and reinforcement learning which is called deep reinforcement learning (DRL) has increased the potential of automation for many decision-making problems like autonomous driving [15]- a sequential decision-making problem which was previously intractable. In this work, we introduce model-free and simulation based learning of DM control policy by adopting state-of-the-art Deep Reinforcement Learning (DRL) approach. Simulating airside traffic in a representative airport environment, it is demonstrated that DRL approach is able to learn, over simulated scenarios, an effective DM policy to contain congestion on an airside network. As a result, taxi delays and average taxi times are reduced significantly without excessive gate holding delays. Concretely, contributions of this work are as follows:

1) To the best of our knowledge, this is the first study to pose the DM problem of assigning TSATs in a DRL framework. In the proposed DRL framework, all departure aircraft are considered as agents, who share a common decentralized DM policy. As aircraft density and fleet size vary dynamically on airside surface, this approach allows for flexibility and scalability, as the learnt policy can be duplicated onto any number of agents.
2) An airside simulator is developed for the purpose of this study that can support the training of DRL algorithms. The simulation environment allows to capture stochastic interactions between agents which give rise to airside environment uncertainty. The novel state observation features (local and global), in the proposed DRL framework, help to learn an effective DM policy under uncertainty.
3) This is the first study to apply Proximal Policy Optimization (PPO; a DRL algorithm) to solve DM problem with high-dimensional state and action spaces. The learnt decentralised agent metering policy is intelligent enough to estimate and minimize potential conflicts with other agent trajectories.

II. Simulation Environment

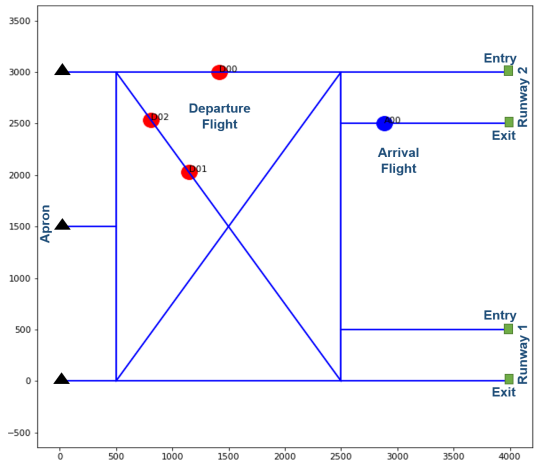

Fig. 2: Airside network layout for training and evaluation of DM policy (dimensions are in meters).

There are two key requirements for designing an airport simulator. First, DLR algorithms are sample inefficient and require training over many episodes before convergence. To assist DM agent training in feasible times, the simulator should therefore be capable of computing desired statistics after generating (hundreds of) scenarios, with low runtime complexity and overhead. Second, the simulator is intended to be used for generating real airport environment scenarios in future studies. Thus, the simulator should be easily configurable to adapt to different airport environments seamlessly with little re-engineering effort. Existing airport simulators were found lacking to meet both these key requirements and thus we decided to develop an airport simulator for training DRL algorithms. To simulate airside traffic, underlying airside network, traffic flow and episode generation mechanisms are required to be defined. Airside network is needed to constrain aircraft movements along the taxiway. Traffic flow mechanisms simulate intended aircraft movements, from respective source to sink nodes, based on generated flight plans in the episode scenario.

A. Airside Network

The airside network consists of three distinct regions - apron, taxiway and runway. In Fig. 2, apron nodes (depicted as black triangles) act as source and sink nodes for departures and arrivals respectively. On the other hand, runway exit and entry nodes (shown as green squares) act as sink and source nodes for departures and arrivals respectively. The (blue) edges depict the taxiway links which are connected at intersections. Exact geometries of apron and runways have not been considered in the simulations. In this work, it is assumed that apron has sufficient gates and capacity to service all departures and arrivals at any given time.

B. Traffic Flow

The simulator consists of four sub-managing systems: departure, surface, runway and arrival managers, that simulate airside traffic based on an episode scenario (flight plan). In flight plan, Target Off-block Time (TOBT; for departures) and Estimated Landing Times (ELDT; for arrivals) are generated randomly.

1) Departure Manager: It controls aircraft release from apron area into the taxiway network.
2) Surface Manager: It controls aircraft movements on the taxiway network. In this work, aircraft must maintain a distance of 200 meters with other aircraft to avoid loss of separation. Aircraft can reach a maximum free flow speed of $10m/s$ and maximum acceleration/deceleration of $3m/s^2$.
3) Runway manager: It controls aircraft occupancy at both runways conditioned on runway occupancy time (ROT) i.e. time period for which an aircraft holds a runway which prevents other aircraft to enter/use it. In this study, we have assumed a constant ROT of one minute.
4) Arrival manager: It controls entry of each arrival aircraft onto the runways at its expected landing time (ELDT). When arrivals and departures compete for runways, arrivals are given preference.

C. Episode scenario generation

1) Fleet mix: The simulation environment can flexibly generate scenarios with different fleet mix and size. As we intend to perform tactical DM, it is assumed that all aircraft have been assigned routes at the time of making DM decisions.
2) Deadlocks: Owing to the limited taxi paths, deadlocks happen when aircraft run into head-on conflicts at middle of a taxiway whereupon it is impossible for either aircraft to retract (aircraft don't have reverse gears) or move forward. These scenarios are discarded and not considered for metering policy evaluations later.

III. MDP Framework

Applying DRL requires formulation of the DM problem as a Markov Decision Process (MDP) where the main variables (State,Action,Reward,Next State) are returned at every timestep: (s, a, r, s'). Before casting the DM problem into a DRL framework (refer Fig. 3), it is worthwhile to note here that DM agent is trained on controlling a single flight per episode. These flights are selected randomly and therefore are different in each training episode. The aim of this work is to train a decentralised agent metering policy which can learn to minimize potential conflicts with other agent trajectories.

A. Observation space

The features are extracted at both flight (local observations) and airport (global observations) levels. These state features are detailed in Table I and can be classified in three broad categories as follows.

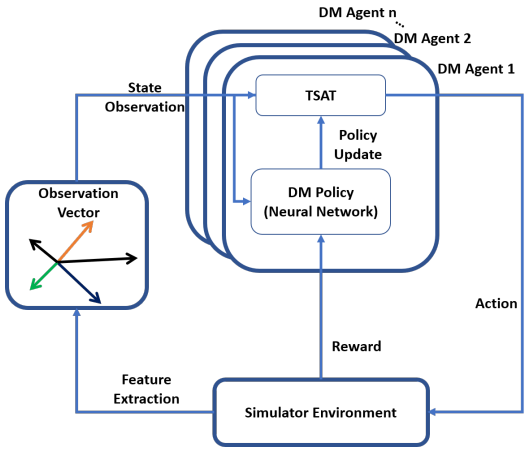

Fig. 3: Proposed framework for DRL approach to the DM problem. State observation features are extracted from simulation environment and are inputted to DM policy. The policy outputs an action (holding or releasing departure aircraft). Based on step and cumulative reward signals, a single agent DM policy is trained to pushback departures to maximize rewards (minimize taxi delays). Once trained, the DM policy is copied to all agents (departure aircraft) for evaluation.

1) Flight plan features: These features are intended to abstract each flight's schedule awareness in both spatial and temporal dimensions. These state features capture the controlled aircraft source and destination locations, intended travel distance and the time already spent at the gate, waiting for pushback clearance.

2) Surface congestion features: These features are intended to abstract both local and airport wide congestion. These state features capture the co-occupation of the route, by other aircraft, that the controlled aircraft intends to traverse, number of aircraft active on the taxiway network and those queuing up before the runways before take-off. It also captures the average moving velocity on the taxiway network- as an indication of surface congestion levels.

3) Airport traffic demand features: These features are intended to abstract expected traffic demand at the airport surface. Based on schedule, it extracts how many arrivals and departures are likely to use the taxiway network in next ten minutes.

B. Action space

At TOBT, a departure flight becomes active. The DM agent has to decide whether to 'hold' or 'release' aircraft which is ready for pushback. If the 'hold' action is chosen, the aircraft pushback decision is delayed until next time step. This process is repeated until the 'release' action is chosen. Then it follows the pre-determined taxi path constrained by the presence of surrounding traffic (other aircraft) on airside network (refer sections II-B and II-C).

C. Reward structure

For every time step t, that a departure is held (waits) at the gate after TOBT, DM agent receives a penalty $r_t =$

Type	Feature	Description
Flight plan	Gate	Gate from which the aircraft is to pushback.
	Runway	Assigned runway, which the departure aircraft shall be heading towards.
	Taxi Distance	Taxi out distance, based on the assigned taxi route trajectory, between gate and runway.
	Wait time	Time elapsed at gate, after TOBT, while the departure aircraft waits for pushback clearance.
Surface congestion	Route occupancy	Number of other aircraft taxiing on the assigned route.
	Co-taxiing aircraft	Total aircraft active on the taxiway network.
	Runway queue length	Number of aircraft queuing at each runway.
	Taxi Speed	Average taxiing speed of all aircraft (arrivals and departures) active on the surface.
Airport demand	Scheduled departures.	Number of departures scheduled in the next 10 minutes.
	Scheduled arrivals.	Number of arrivals scheduled in the next 10 minutes.

I. Classification of features used to encode state space

$-\zeta$ where ζ is a constant. After the departure is released (pushed back), agent receives $r_t = -2\zeta$ for every time step t the departure spends on taxiway network. Once the departure reaches runway and is ready to take-off, DM agent receives a final positive reinforcement (reward) M- based on $TOT_{unimpeded}$. $TOT_{unimpeded}$ is the unimpeded taxi out time of the departure aircraft which is computed by dividing the assigned taxi path length with the preferred (free flow) aircraft speed. At the end of each episode, the return R of an agent is equal to the sum of rewards over all time steps as follows.

$$\begin{aligned} R &= \sum_{t=1}^{T-1} r_t + M \\ &= \underbrace{(\sum_{t=1}^{p} -\zeta t)}_{\text{gate hold delay}} + \underbrace{(\sum_{t=p+1}^{T-1} -2\zeta t) + M}_{\text{taxi out delay}} \end{aligned} \tag{1}$$

where p is the time step at which aircraft is pushed back from the gate and T is the final time step at which the aircraft starts lining up at the runway for take-off.

$$M = 2\zeta TOT_{unimpeded} \tag{2}$$

In essence, this reward structure encourages the agent to assign TSATs to minimize overall gate hold times and taxi delays. However, since departure delays on taxiway are less desirable than hold delays on gate, additional taxi time (in excess of $TOT_{unimpeded}$) is penalised twice as much as the total gate hold time.

D. Policy

Proximal Policy Optimization (PPO)- a model-free algorithm, is used to train DM agent policy. While the learnt policy should quickly adapt to environment uncertainty, it should not become unstable. PPO is chosen as it strikes a balance between sample complexity; ease of tuning and minimizes the cost function while ensuring the deviation from the previous policy is relatively small. PPO is effectively able to learn (refer section V) the functional mapping between state, action and rewards. Thus, the trained agent does not have to solve every problem from scratch, which may greatly reduce the response times- an indispensable requirement for tactical DM.

IV. Experiments

To find optimal DM policy, we have used PPO implementation in [16] to train an agent. The policy is trained over varying fleet mix (arrival to departure ratio) and one departure is controlled in every scenario. However, the fleet size is fixed to 10 aircraft- which is experimentally found to strike a good balance between computation times (which increase with increase in number of aircraft) and generating enough congestion to learn conflict avoidance with other aircraft trajectories. It is observed while training that an episode may continue indefinitely if a deadlock occurs between two aircraft (refer section II-C). It is worthwhile to note that typically number of such deadlock episodes observed were less than 1% ($\approx$ 0.8%) of the total evaluation episodes. Nonetheless, to avoid such a situation, a maximum step size of 3000 (50 minutes) has been fixed for experiments. If an episode fails to terminate, it is not evaluated. Table II details training parameters used to train the PPO algorithm.

To evaluate the obtained DM policy, we compute taxi delays and average taxi times, with and without DM, on exact same unseen scenarios (refer Fig. 4 and Fig. 5). For these experiments, airside traffic consisted of 34 aircraft in a 50 minute (max.) scenario (corresponding to average traffic density of 40 aircraft/hour). This is equivalent to the average traffic density observed at mid to large sized airports like Singapore Changi [7], Charles de Gaulle (Paris; CDG) and Charlotte Douglas International (North Carolina; CLT). However, in reality, airside traffic density varies throughout the day. For instance, even when the traffic density at CDG and CLT averages equally around 40 aircraft/hour, departure demand is significantly banked at CLT compared to CDG,

Parameters	Value
Training Iterations	3.5e+5
Env. steps per update	128
Mini batches per update	4
ANN architecture	Multilayer Perceptron
Hidden Layers	64 X 64
Clipping coefficient	0.2
Value function coefficient	0.5
Gamma	0.99
Gradient clipping (max.)	0.5
Learning rate	2.5e-4
Reward Coefficient (ζ)	1

II. Parameters for PPO training to find optimal DM policy.

resulting in periods of higher demand-capacity imbalance, leading to the formation of larger queues [10]. To evaluate performance of the proposed DRL approach in different environment complexities (other than the one it is trained on) we run experiments with varying fleet-sizes (refer Fig. 6). Moreover, since excessive gate holds by departures may disrupt gate allocation plans for incoming arrivals (when arrival and departure are allocated the same gate), we also compute the average gate holding times due to metering in all scenarios (refer Fig. 7). For each figure, from Fig. 4 to Fig. 7, we run 250 scenarios to obtain accurate delay, average taxi and gate holding time distributions.

V. Policy Evaluation Results

The DM agent's policy stabilises after around 350K training episodes when ζ is fixed to be 1 in reward structure (refer section III-C). Fig. 4 shows that total taxi delays reduce significantly due to learnt DM policy. As is evident from the figure, the delay distribution shifts left towards lower delay values. Due to DM agent policy, the median delay drops approximately from 3500 sec. to 2600 sec.- a reduction of about 25% in taxi delays in a scenario with traffic density of 40 aircraft/hour.

	Mean	Std.Dev.	Median
Non Metering	3495	1283	3315
Metering	2631	1286	2489

III. Total taxi delay (seconds) with and without metering.

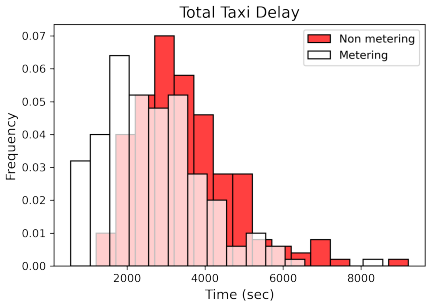

Fig. 4: Reduction in total taxi delay due to DM policy.

Due to lesser taxi delays, average time that an aircraft spends taxiing also reduces. Fig. 5 shows that the taxi time distribution shifts leftwards. Due to DM policy, the mean value drops from 725 to 675 seconds- a taxi time saving of 50 seconds per flight. In a typical day, a medium sized airport serves 1000 flights. Thus, these savings scale to approx. 14 hours (50,000 seconds) of overall taxi time (and corresponding fuel burn) savings in a day.

	Mean	Std.Dev.	Median
Non Metering	725	81	710
Metering	675	84	667

IV. Average taxi time (seconds) with and without metering.

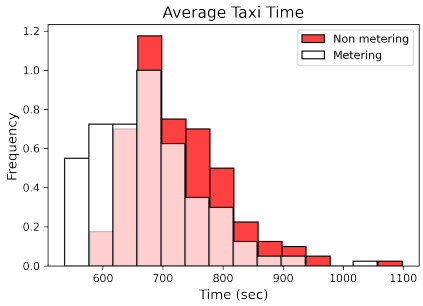

Fig. 5: Reduction in average taxi time due to DM policy.

Figures 6 shows that as traffic density on the airside network increases, average per flight taxi time savings show improvements. When airside surface traffic increases from 30 to 70 aircraft in a 50 minute (max.) scenario, average taxi time savings increase from 23 to 247 seconds (refer Table V).

Fleet Size	Taxi Time Saving	Gate Holding
30	23	431
50	143	423
70	247	422

V. Effect of traffic density (unit: number of aircraft) on average taxi time savings and additional gate holding time (mean value in seconds).

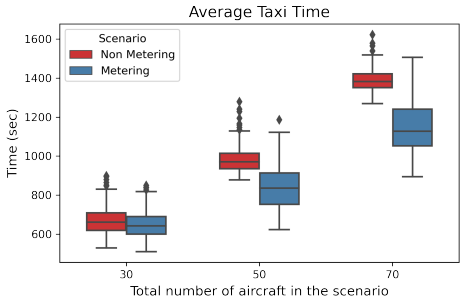

Fig. 6: Difference in average taxi time, between metered and non-metered scenarios, with increase in scenario fleet size.

Fig. 7 shows that average gate hold times remain stable around 420-430 seconds with increase in scenario fleet size.

This happens, potentially because, under increased congestion, increased gate holdings can not earn proportional taxi time savings and agent eventually chooses similar gate holds (as under lower density environments) at the cost of increased taxi times (refer Fig. 6). In few exceptional scenarios, the average gate hold times reach around 800 seconds (see outliers in scenario with fleet size = 30). Excessive gate holds by departures may disrupt gate allocation plans for incoming arrivals (when arrival and departure are allocated the same gate). Although it varies from country to country, most airports expect actual departure time to be within +/- 15 minutes (900 sec.) from scheduled departure times for on-time performance- a Key Performance Indicator. Thus, future research shall aim to restrict gate holding times more explicitly by penalizing gate holds harshly beyond a certain threshold.

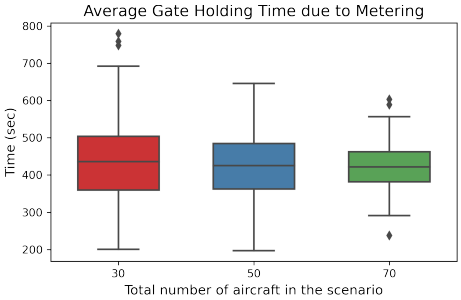

Fig. 7: Average gate holds due to DM policy. With increase in fleet size, the gate holding times remain relatively stable.

VI. Conclusion

In this work, we proposed a DRL approach for controlling pushback times (TSATs) of departures. We employed PPO algorithm to train DRL agent into finding an effective DM policy. Since multiple agents are designed to share a common, single-agent policy, the final learned policy can be copied onto any number of agents. Simulating airside surface traffic in a representative airport environment, it is shown that DRL approach is able to learn, on simulated scenarios, an effective DM policy to contain congestion on an airport airside. As a result, taxi delays and average taxi times are reduced significantly. Through extensive simulation experiments, it is demonstrated that the final trained policy naturally scales to various fleet sizes. With increase in traffic density, taxi time savings obtained by the DM policy improves without significant increase in average gate holding times.

As a next step, this work shall be adopted for Singapore Changi Airport, using surveillance (Advanced Surface Movement Guidance and Control System (A-SMGCS)) and actual airport network data. Moreover, in future work, effect of metering on interactions between apron, taxiways and runways will also be evaluated.

Acknowledgement

This research is supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore, under the Aviation Transformation Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore.

References

[1] H. Idris, N. Shen, A. Saraf, J. Bertino, and S. Zelinski, "Comparison of different control schemes for strategic departure metering," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. IEEE, 2016, pp. 1–13.

[2] H. Chen and S. Solak, "Lower cost departures for airlines: Optimal policies under departure metering," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 531–546, Feb. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X19307004

[3] H. Ali, Y. Guleria, S. Alam, and M. Schultz, "A Passenger-Centric Model for Reducing Missed Connections at Low Cost Airports With Gates Reassignment," *IEEE Access*, vol. 7, pp. 179 429–179 444, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8918374/

[4] H. Ali, Y. Guleria, S. Alam, V. N. Duong, and M. Schultz, "Impact of stochastic delays, turnaround time and connection time on missed connections at low cost airports," in *Proc. 13th USA/Eur. Air Traffic Manage. R&D Seminar*, 2019.

[5] I. Simaiakis and H. Balakrishnan, "Impact of congestion on taxi times, fuel burn, and emissions at major airports," *Transportation research record*, vol. 2184, no. 1, pp. 22–30, 2010.

[6] E. Feron, R. J. Hansman, A. R. Odoni, R. B. Cots, B. Delcaire, X. Feng, W. D. Hall, H. R. Idris, A. Muharremoglu, and N. Pujet, "The departure planner: A conceptual discussion. white paper, massachusetts institute of technology," *International Center for Air Transportation*, 1997.

[7] H. Ali, R. Delair, D.-T. Pham, S. Alam, and M. Schultz, "Dynamic hot spot prediction by learning spatial-temporal utilization of taxiway intersections," in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*. IEEE, 2020, pp. 1–10.

[8] M. C. R. Murça, "A robust optimization approach for airport departure metering under uncertain taxi-out time predictions," *Aerospace Science and Technology*, vol. 68, pp. 269–277, Sep. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1270963817308854

[9] L. Yang, S. Yin, K. Han, J. Haddad, and M. Hu, "Fundamental diagrams of airport surface traffic: Models and applications," *Transportation Research Part B: Methodological*, vol. 106, pp. 29–51, Dec. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0191261517303843

[10] S. Badrinath, H. Balakrishnan, J. Ma, and D. Delahaye, "Comparative analysis of departure metering at united states and european airports," *Journal of Air Transportation*, pp. 1–12, 2020.

[11] I. Simaiakis, M. Sandberg, and H. Balakrishnan, "Dynamic Control of Airport Departures: Algorithm Development and Field Evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 285–295, Feb. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6589157/

[12] S. Badrinath, H. Balakrishnan, E. Joback, and T. G. Reynolds, "Impact of off-block time uncertainty on the control of airport surface operations," *Transportation Science*, vol. 54, no. 4, pp. 920–943, 2020.

[13] R. Mori, "Evaluation of departure pushback time assignment considering uncertainty using real operational data," *9th SESAR Innovation Days*, 2019.

[14] ——, "Optimal pushback time with existing uncertainties at busy airport," in *Proceedings of 29th Congress of the ICAS, St. Petersburg*, 2014.

[15] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[16] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.