

Runway Configuration Management with Offline Reinforcement Learning

Milad Memarzadeh*, Tejas Puranik[†], Jarrett Battistini[‡], Krishna Kalyanam, Wes Ryan
NASA Ames Research Center, Moffett Field, CA 94035, USA

I. Extended Abstract

Runway configuration management (RCM) is a challenging task, and it affects the efficiency of the National Airspace System (NAS) and airport surface operations significantly. Each airport, depending on the geometry, capacity, local climate patterns, etc. has multiple configurations for the runway usage for arriving and departing flights. Many factors such as the incoming/outgoing traffic load, wind direction and speed, convective weather, cloud ceiling and other environmental factors might affect the choice of a runway configuration at any point in time. However, other factors such as safety measures and regulations, noise abatement, capacity of each configuration, and preference of the air traffic controllers (ATCs) can also play a significant role in selecting the configuration. A sub-optimal selection of the runway configuration, or delay in making configuration changes might result in significant increase in taxi times for aircraft on the surface of the airport, fuel and energy use of the aircraft, and maintenance costs. It can also lead to safety concerns, such as an aircraft performing one or more go-arounds before being able to land. All these factors make RCM an extremely important and challenging decision-making process for the ATCs.

The current state of practice sets the runway configuration by the ATCs based on relevant information available at the time including weather, traffic, noise abatement, safety bounds, etc. This makes the decision-making process subjective based on the accuracy of the available information and the bias in human decision making. Unfortunately, this approach yields poor results (e.g., significant delays) if the predicted outcomes are uncertain and their relative impact is not well understood. This is especially evident when the uncertainty increases the size of possible predicted outcomes (combinatorial explosion in possible scenarios) that cannot be handled by human reasoning. On the other hand, an automated approach based on machine intelligence can make use of historical data and search through all (or significant amount of) possible scenarios under uncertainty and make well-informed decisions.

AI/ML has been used to address the RCM problem. One popular area uses different variants of model-based Reinforcement Learning (RL), such as discrete choice modeling [1], dynamic programming [2], and its combination with queue modeling [3, 4] to model the dynamics of the surface operations at an airport and use the simulation to learn the near-optimal policy for the runway configuration. This approach is interpretable (due to learning of the dynamics model for both traffic and weather) and is guaranteed to provide near-optimal solution. However, its performance is dependent on characterizing and building an accurate simulation (i.e., dynamics model) that is used for learning the policy. Any error/inaccuracy in the simulation model that is caused by noise in the system or over-simplification (such as assuming simplified weather dynamics models [1]) can result in poor performance in the operational setting and/or safety concerns. Furthermore, since the dynamics model is different from airport to airport, generalization of this framework to different airports across the nation is challenging. A potential solution is to rely on an online model-free RL approach such as Monte Carlo Tree Search (MCTS) [5] for learning a good policy, however, such online approaches still require an accurate simulation environment to interact with. This interaction is impractical in safety-critical problems, either because data collection is expensive and/or dangerous. For example, a bad decision in the RCM problem can result in a safety incident involving landing or take-off of a commercial aircraft. As a result, they are not applicable to safety-critical systems such as the RCM problem (or autonomous driving) and majority of the literature in online RL is focused on non-critical applications such as gaming [6].

As a result, majority of the more recent literature is focused on data-driven (or sometimes called model-free) approaches to tackle the RCM problem. One area of work uses supervised machine learning to predict the best choice of runway configuration given all the independent factors such as weather and future traffic [7, 8]. Although these approaches show great accuracy in predicting the runway configuration, such prediction is not backed up by any evidence that this would decrease the transit times or would alleviate a safety concern. The reason is that supervised learning

*Affiliated with USRA, Corresponding Author: milad.memarzadeh@nasa.gov

[†]Affiliated with USRA.

[‡]Affiliated with Texas A&M University.

does not build a utility function and is not predicting the runway configuration to optimize the operational utility. Furthermore, supervised learning assumes that each data instance is i.i.d. (independent and identically distributed). In the RCM problem, each decision influences the future events, which in turn influence future decisions, as a result the assumption about the data being i.i.d. is violated.

In this paper, we propose to deploy an offline RL methodology to address the RCM problem. Offline RL combines reinforcement learning (a key framework for any sequential decision-making problem) with data-driven machine learning. More importantly, it removes the need for an online interaction with the system for data collection, which limited the applicability of online RL methods to safety-critical problems [9]. Although offline RL has been the focus of study in many fields (e.g., autonomous driving, healthcare, robotics), there are fundamental challenges to their deployment that classical methods [10] fail to address. One such challenge is the offline nature of the algorithm, which means that there is no exploration involved in learning the policy. Another major challenge is the distributional shift. This happens when the behavioral policy (an unknown policy that the historical data are collected based on) is different from the policy that is learned by the algorithm using that data. For example, if a state (a specific scenario involving traffic and weather) is not well represented in the historical data, then the policy learned for that state is overly optimistic and most probably wrong. In the online RL, it is easy to correct such errors by interacting with the system and receiving feedback, however, in the classical offline RL, there is no such mechanism for error correction. As a result, majority of the state-of-the-art offline RL algorithms address this issue by either constraining the optimization problem (learning the policy) [11], deploying an ensemble approach [12], or performing conservative updates in the learning scheme to avoid the excessive distribution shift from the behavioral policy [13].

We will deploy a state-of-the-art offline RL algorithm, called Conservative Q-Learning (CQL) [13], to the RCM problem and compare its performance to a classical counterpart, a popular Deep Q-Network (DQN) [6], in an offline setting. For validation, we will use Charlotte Douglas International Airport (CLT) as a case study and will utilize NASA’s Sherlock Data Warehouse, as well as FAA’s SWIM database to obtain traffic and weather data. Once the model is validated for CLT, we will evaluate its generalization and applicability to multiple other airports across the NAS.

References

- [1] Avery, J., and Balakrishnan, H., “Data-Driven Modeling and Prediction of the Process for Selecting Runway Configurations,” *Transportation Research Record*, Vol. 2600, 2016. <https://doi.org/10.3141/2600-01>.
- [2] Li, L., Clarke, J.-P., Chien, H.-H., and Melconian, T., “A probabilistic decision-making model for runway configuration planning under stochastic wind conditions,” *IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009. <https://doi.org/10.1109/DASC.2009.5347528>.
- [3] Jacquillat, A., Odoni, A. R., and Webster, M. D., “Dynamic Control of Runway Configurations and of Arrival and Departure Service Rates at JFK Airport Under Stochastic Queue Conditions,” *Transportation Science*, Vol. 51, No. 1, 2016, pp. 155–176. <https://doi.org/10.1287/trsc.2015.0644>.
- [4] Badrinath, S., Li, M. Z., and Balakrishnan, H., “Integrated Surface–Airspace Model of Airport Departures,” *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 5, 2019, pp. 1049–1063.
- [5] Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S., “A Survey of Monte Carlo Tree Search Methods,” *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, 2012. <https://doi.org/10.1109/TCIAIG.2012.2186810>.
- [6] Mnih, V., Kavukcuoglu, K., Silver, D., and et al., “Human-level control through deep reinforcement learning,” *Nature*, Vol. 518, 2015, pp. 529–533.
- [7] Khater, S., Rebollo, J., and Coupe, W. J., “A Recursive Multi-step Machine Learning Approach for Airport Configuration Prediction,” *AIAA Aviation Forum*, 2021. <https://doi.org/10.2514/6.2021-2406>.
- [8] Churchill, A., Coupe, W. J., and Jung, Y. C., “Predicting Arrival and Departure Runway Assignments with Machine Learning,” *AIAA Aviation Forum*, 2021. <https://doi.org/10.2514/6.2021-2400>.
- [9] Levine, S., Kumar, A., Tucker, G., and Fu, J., “Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems,” *Arxiv*, 2020. URL <https://arxiv.org/abs/2005.01643>.
- [10] Sutton, R. S., and Barto, A. G., “Introduction to Reinforcement Learning,” *MIT Press, Cambridge, MA*, 2018, 2018.

- [11] Fujimoto, S., Meger, D., and Precup, D., “Off-Policy Deep Reinforcement Learning without Exploration,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2019, pp. 2052–2062.
- [12] Agarwal, R., Schuurmans, D., and Norouzi, M., “An Optimistic Perspective on Offline Reinforcement Learning,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2020, pp. 104–114.
- [13] Kumar, A., Zhou, A., Tucker, G., and Levine, S., “Conservative Q-Learning for Offline Reinforcement Learning,” *Arxiv*, 2020. URL <https://arxiv.org/abs/2006.04779>.