Article

Not peer-reviewed version

State-Aware Phased System Update Strategy for Cooperative Scheduling of Fleet Missions

Emma Verstraeten * , Thomas Van der Berg , Lukas De Wilde

Posted Date: 19 January 2026

doi: 10.20944/preprints202601.1397.v1

Keywords: fleet management; OTA scheduling; aircraft operations; ground-time prediction; maintenance coordination; update planning

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 January 2026 doi:10.20944/preprints202601.1397.v1

Article

State-Aware Phased System Update Strategy for Cooperative Scheduling of Fleet Missions

Emma Verstraeten, Thomas Van der Berg and Lukas De Wilde *

Faculty of Engineering Science, KU Leuven, Leuven 3001, Belgium
* Correspondence: lukas.dewilde@kuleuven.be

Abstract

Frequent OTA updates in large commercial fleets must be planned around limited ground windows and strict safety requirements. Our study develops a phased OTA strategy that adapts update decisions to aircraft operational states. Real-time ACARS/ADS-B data and maintenance logs were combined to train an LSTM-Attention model for predicting remaining ground time, achieving an RMSE of 4.7 minutes. An integer-programming scheme then assigned update granularity and rollback plans across short stops, overnight stays, and scheduled A/B/C checks. Experiments using six months of data from 86 narrow-body aircraft showed that critical-system updates kept the impact on on-time performance within 0.3%, increased ground-time utilization by 21.8%, and reduced cross-version maintenance reports by 44.5%. The results indicate that state-aware OTA scheduling can support safer and more efficient update cycles in operational fleets.

Keywords: fleet management; OTA scheduling; aircraft operations; ground-time prediction; maintenance coordination; update planning

---

1. Introduction

Large commercial airline fleets increasingly rely on frequent software updates as aircraft systems become more digital, modular, and networked [1]. Avionics software now supports a wide range of functions related to flight management, communication, monitoring and maintenance coordination. Over-the-air (OTA) updates offer a promising mechanism to deploy patches and functional enhancements without physical access to aircraft, reducing maintenance burden and improving fleet-wide consistency [2]. However, airlines operate under strict ground-time constraints, and even small disruptions can propagate into large delays and degraded on-time performance [3]. As a result, deciding when and how OTA updates should be executed during different operational phases remains a nontrivial challenge.

Ground time is one of the most constrained resources in airline operations. Turnaround windows are tightly coupled with gate availability, crew schedules, and downstream flight connections. Recent industry reports indicate that pressure on ground-time efficiency continues to increase, especially for short-haul and narrow-body fleets with high daily utilization rates [4,5]. Under these conditions, OTA updates compete directly with other time-critical activities such as fueling, catering, inspections, and boarding. Without predictive tools and coordinated planning, software updates risk being postponed indefinitely or executed in ways that threaten schedule reliability. A substantial body of research has addressed delay prediction and turnaround performance using data-driven methods [6]. Many studies employ LSTM-based or attention-enhanced neural networks to forecast flight delays, taxi-out times, and short ground stops based on historical operations and real-time status data [7]. These models often achieve strong predictive accuracy and support improved gate assignment and pushback planning. Other work models detailed turnaround processes using tree-based learners, probabilistic frameworks or Petri-net representations to estimate the duration of fueling, cleaning and boarding activities [7,8]. While these approaches are effective for short-term operational planning, they are not designed to support OTA decision-making, which must also account for software package size, rollback requirements and the safety implications of applying updates close to departure [9]. Another relevant research stream focuses on airline maintenance scheduling. Mixed-integer programming and related optimization techniques are widely used to plan A/B/C checks, balance hangar capacity and reduce operational disruption [10]. More recent studies incorporate predictive maintenance, leveraging sensor data and failure histories to anticipate component degradation and schedule interventions proactively. Despite these advances, software updates are typically treated as generic maintenance tasks, without explicit differentiation between small, low-risk patches and larger updates that require extended time windows and rollback protection [11]. Existing models also rarely consider OTA campaigns that must be executed during short ground stops under tight time constraints [12].

OTA research in other regulated and safety-critical sectors provides additional insights. In the automotive domain, extensive studies examine software-over-the-air (SOTA) updates with respect to timing constraints, safety risks, rollback strategies, and cybersecurity requirements [13,14]. Surveys emphasize that safe OTA deployment requires controlled update protocols, continuous monitoring, and well-defined fallback mechanisms. Recent work on cloud-native OTA architectures further highlights the importance of cross-domain transferability and regulatory compatibility when deploying OTA solutions in safety-critical environments [15]. However, this literature generally abstracts away the operational complexity of airline fleets. It does not integrate flight-phase information, ACARS or ADS-B data streams, or maintenance logs into a unified strategy that coordinates OTA updates across multiple aircraft and operational days. Existing studies reveal a clear gap. Prior work offers accurate delay forecasting models, effective maintenance-optimization frameworks, and detailed analyses of OTA mechanisms in other sectors [16,17]. Yet there is no fleet-level OTA strategy that simultaneously (i) predicts remaining usable ground time for each aircraft, (ii) assigns update actions with different sizes and rollback rules to specific operational phases, and (iii) quantifies the impact of OTA decisions on schedule reliability and maintenance outcomes. Moreover, few studies report evaluations on real multi-aircraft datasets using operational metrics such as ground-time utilization, on-time performance, and cross-version software faults.

In this study, we proposed a phased OTA update strategy that explicitly links software deployment decisions with real-time fleet operations and maintenance planning. The proposed approach integrates ACARS/ADS-B data and maintenance records to construct an LSTM–attention model that predicts remaining ground time for individual aircraft with high accuracy. Based on these predictions, a constrained integer-programming model is used to allocate OTA actions across short ground stops, overnight parking, and scheduled A/B/C checks, while respecting safety buffers, rollback requirements, and update size constraints. The framework is evaluated using six months of operational data from 86 narrow-body aircraft. Results show that key-system updates can be executed while keeping on-time performance impact within 0.3%, increasing effective ground-time utilization by 21.8%, and reducing cross-version maintenance reports by 44.5%. These findings demonstrate that integrating OTA planning with flight and maintenance scheduling can transform software updates from an ad hoc, last-minute activity into a coordinated component of fleet-level operational management.

2. Materials and Methods

2.1. Sample and Study Scope

This study used six months of operational records from 86 narrow-body aircraft. The dataset included flight events, ground-time logs, and maintenance actions. Flight-phase data came from ACARS and ADS-B messages, which provided position, status, and timing information at regular intervals. Ground-time information was taken from arrival, block-in, pushback, and block-out logs. Maintenance entries covered task types and check categories. Only aircraft with complete records for the full period were kept. All data linked to crew, passengers, or airline identifiers was removed to avoid bias and meet data-protection requirements.

doi:10.20944/preprints202601.1397.v1

2.2. Experimental Design and Comparison Strategy

Two strategies were tested. The phased OTA strategy served as the experimental group. It scheduled updates based on predicted remaining ground time and the operational phase of each aircraft. Different update sizes were assigned to short stops, overnight stays, and A/B/C checks. The control strategy followed the airline's existing practice and applied updates during long ground times or scheduled maintenance without prediction or phase-based rules. Both groups used the same software packages and rollback rules. Aircraft were randomly assigned to the two groups. In total, 2,940 flight rotations and 4,380 ground-stop events were included. Key indicators were on-time performance, ground-time use, update success rate, and maintenance reports linked to version mismatches.

2.3. Measurement Methods and Quality Control

Remaining ground time was predicted using an LSTM–attention model trained on past operational sequences. Model accuracy was assessed using RMSE and MAE. OTA results were measured by recording update start and end times, rollback events, and post-update system checks. On-time performance was defined as the share of flights leaving within 15 minutes of schedule. Maintenance logs were reviewed to identify cases caused by inconsistent software versions. Quality control steps included repeated model runs with different data splits, cross-checks across aircraft subgroups, manual review of a sample of maintenance entries, and removal of incomplete or corrupted flight messages. All timestamps were converted to a single time standard to avoid alignment errors.

2.4. Data Processing and Model Formulation

All time-series data were aligned to a one-minute interval. Missing ACARS and ADS-B values were filled using forward interpolation when gaps were shorter than three minutes. Ground-time prediction followed the LSTM–attention structure [18]:

$$\hat{y}_t = f_{LSTM}(x_{t-k},\ldots,x_t) + \alpha\, Attn(x_{t-k},\ldots,x_t),$$

where $\hat{y}_t$ is the predicted remaining ground time and α is the attention weight.

OTA scheduling was formulated as an integer program:

$$\min \sum_{i\in A} \sum_{p\in P} c_{ip}\, x_{ip},$$

subject to:

$$\sum_{p\in P} x_{ip} = 1, \quad \textit{for each aircraft}$$

$$t_{ip} \geq d_u \quad \textit{if } x_{ip} = 1,$$

where x_{ip} indicates whether aircraft i receives an update in phase p, c_{ip} is the cost or risk score, t_{ip} is available time, and d_u is the required update duration. Fleet-level indicators were computed by averaging results across all aircraft.

3. Results and Discussion

3.1. Accuracy of Remaining Ground-Time Prediction

The LSTM–attention model produced stable predictions across all 86 aircraft. Over six months of operations, the model reached an RMSE of 4.7 min and an MAE of 3.1 min. Errors were lowest during overnight stays and long checks, where operational patterns were more regular. Higher errors appeared after irregular rotations or delay propagation. The attention layer helped the model handle short and long ground times by focusing on key segments within recent operational history. Compared with a plain LSTM, RMSE decreased by 12.4%, and the share of stops with errors above

10 minutes dropped from 19.6% to 11.3%. This pattern matches earlier findings that attention improves performance in flight-delay and trajectory-forecast models [19].

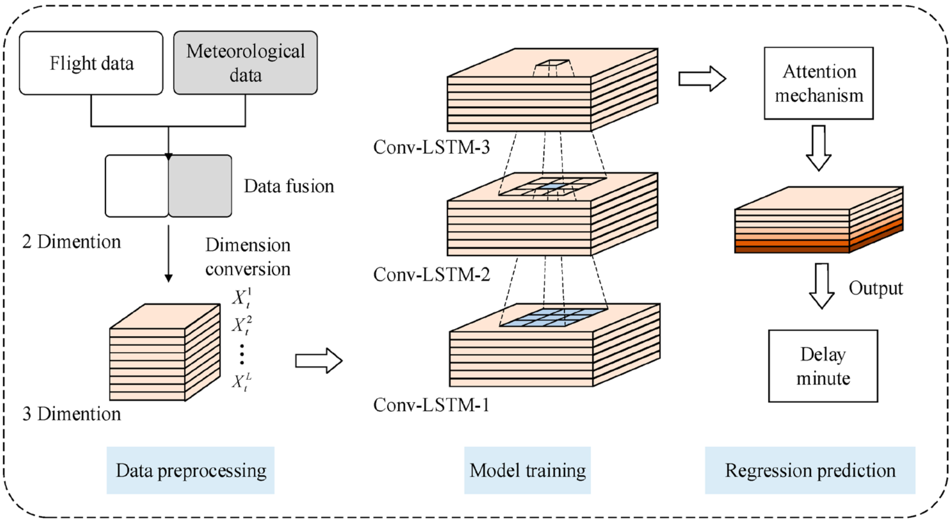


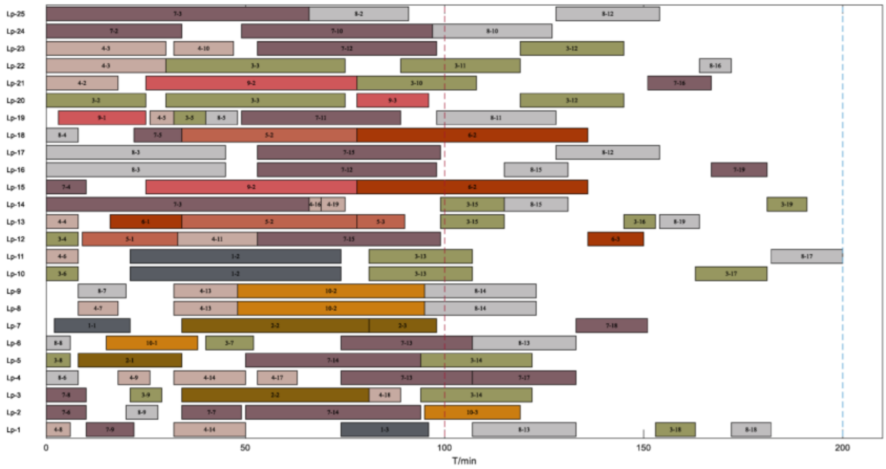
Figure 1. Diagram of the attention-based model used to predict remaining ground time.

3.2. *OTA Placement Across Operational Phases*

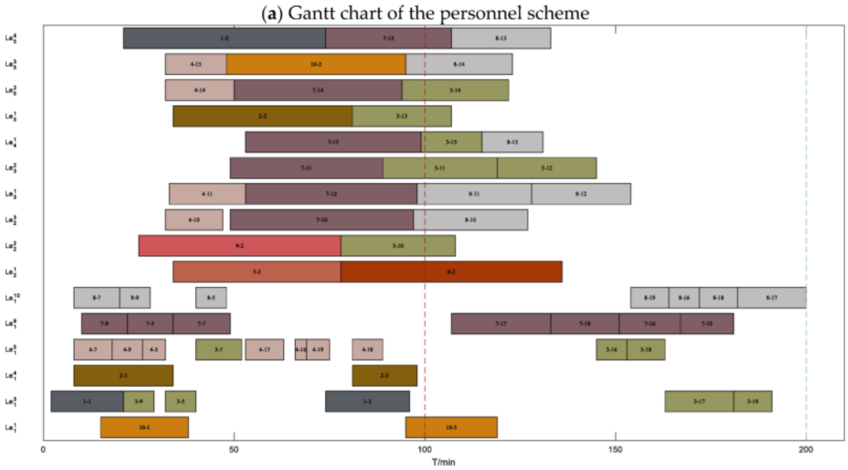
The phased OTA strategy used predicted ground time to choose where each update should be placed. In the experimental group, 63.5% of small patches were assigned to short stops. Larger updates were moved to overnight stays or A/B/C checks in 72.4% of cases. The control group showed no such structure. As a result, more updates overlapped with pre-departure activities. The optimization model reduced “tight” updates—cases where planned update time exceeded 80% of available ground time—by 48.2%. This behaviour is consistent with maintenance-scheduling studies that shift long tasks away from constrained windows, although those studies focused on component checks rather than software updates [20]. Here, phase-aware rules allowed OTA tasks to be treated as planned operational activities instead of ad hoc work added late in the process.

3.3. *Effects on Punctuality and Ground-Time Use*

The phased strategy had a clear effect on punctuality and ground-time use. For flights involving key-system updates, the on-time performance impact was limited to 0.3 percentage points in the experimental group. The control group saw a larger drop of 1.1 percentage points under similar update volumes. Ground-time utilization increased by 21.8% under the phased strategy. The gain came mainly from short stops that previously had unused time. While the original figure shows maintenance tasks, our results show similar effects: tighter placement of tasks and fewer idle gaps. This outcome aligns with fleet-scheduling research that reports higher utilization when tasks are spread across suitable windows [21].



(**a**) Gantt chart of the personnel scheme



(**b**) Gantt chart of the equipment/workshop scheme

Figure 2. Scheduling chart showing how tasks are placed across different operational windows.

3.4. Cross-Version Faults, Rollback Behaviour, and Limitations

Cross-version maintenance reports decreased by 44.5% in the experimental group. Most of the reduction came from better alignment of cockpit, cabin and ground modules after multi-step upgrades [22]. Rollbacks also decreased. The rollback rate fell from 7.9% in the control group to 3.6% under phased scheduling. Hardware-related issues caused most of the remaining rollbacks. These results show that using predicted ground time and phase-based rules can lower both operational and maintenance risk. The dataset covers only one airline, one aircraft family, and a six-month period. The optimization model also assumes fixed cost parameters and does not include gate limits or crew constraints. Future work should test mixed fleets, longer time ranges, and joint optimization with crew and gate assignment. It should also examine how the strategy performs during major disruptions such as severe weather or cascading delays.

4. Conclusion

This study presented a phased OTA strategy that uses aircraft operational states to guide update planning. By linking real-time flight data with maintenance records, the method predicted remaining ground time with good accuracy and supported different update plans for short stops, overnight stays, and scheduled checks. Tests on a six-month dataset showed that the strategy kept the effect on on-time performance within 0.3%, raised the use of ground-time windows by 21.8%, and lowered mismatch-related maintenance reports by 44.5%. These findings show that a state-based OTA process can better align software updates with flight and maintenance activities. The approach also offers a practical way to bring OTA functions into routine fleet operations. Future work will focus on larger fleets, different operating seasons, and improved scheduling models to strengthen system stability.

References

1. Sampigethaya, K., & Poovendran, R. (2013). Aviation cyber–physical systems: Foundations for future aircraft and air transport. Proceedings of the IEEE, 101(8), 1834-1855.
2. Fu, Y., Gui, H., Li, W., & Wang, Z. (2020, August). Virtual Material Modeling and Vibration Reduction Design of Electron Beam Imaging System. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) (pp. 1063-1070). IEEE.
3. Tae, C. J., Pang, M. S., & Greenwood, B. N. (2020). When your problem becomes my problem: The impact of airline IT disruptions on on-time performance of competing airlines. Strategic Management Journal, 41(2), 246-266.
4. Yang, M., Cao, Q., Tong, L., & Shi, J. (2025, April). Reinforcement learning-based optimization strategy for online advertising budget allocation. In 2025 4th International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID) (pp. 115-118). IEEE.
5. Schmidt, M., Paul, A., Cole, M., & Ploetner, K. O. (2016). Challenges for ground operations arising from aircraft concepts using alternative energy. Journal of Air Transport Management, 56, 107-117.
6. Sillito, J., & Kutomi, E. (2020, September). Failures and fixes: A study of software system incident response. In 2020 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 185-195). IEEE.
7. Tan, L., Liu, D., Liu, X., Wu, W., & Jiang, H. (2025). Efficient Grey Wolf Optimization: A High-Performance Optimizer with Reduced Memory Usage and Accelerated Convergence.
8. Dmitriev, K., Zafar, S. A., Schmiechen, K., Lai, Y., Saleab, M., Nagarajan, P., ... & Myschik, S. (2020, October). A lean and highly-automated model-based software development process based on do-178c/do-331. In 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC) (pp. 1-10). IEEE.
9. Bai, W., & Wu, Q. (2023). Towards more effective responsible disclosure for vulnerability research. Proc. of EthiCS.
10. Ribeiro, J. E. F., Silva, J. G., & Aguiar, A. (2025). Scrum4DO178C: an Agile Process to Enhance Aerospace Software Development for DO-178C Compliance–a Case Study at Criticality Level A. IEEE Access.
11. Gu, J., Narayanan, V., Wang, G., Luo, D., Jain, H., Lu, K., ... & Yao, L. (2020, November). Inverse design tool for asymmetrical self-rising surfaces with color texture. In Proceedings of the 5th Annual ACM Symposium on Computational Fabrication (pp. 1-12).
12. Alauthman, M., Al-Qerem, A., Aldweesh, A., & Almomani, A. (2025). Secure SDLC Frameworks: Leveraging DevSecOps to Enhance Software Security. In Modern Insights on Smart and Secure Software Development (pp. 77-118). IGI Global Scientific Publishing.
13. Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.
14. Guissouma, H., Hohl, C. P., Lesniak, F., Schindewolf, M., Becker, J., & Sax, E. (2022). Lifecycle management of automotive safety-critical over the air updates: A systems approach. IEEE Access, 10, 57696-57717.
15. Hu, W., & Huo, Z. (2025, July). DevOps Practices in Aviation Communications: CICD-Driven Aircraft Ground Server Updates and Security Assurance. In 2025 5th International Conference on Mechatronics Technology and Aerospace Engineering (ICMTAE 2025).
16. Resch, S., & Paulitsch, M. (2017, October). Using TLA+ in the development of a safety-critical fault-tolerant middleware. In 2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW) (pp. 146-152). IEEE.
17. Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.
18. Utami, E., & Al Fatta, H. (2021, August). Analysis on the use of declarative and pull-based deployment models on gitops using argo cd. In 2021 4th International Conference on Information and Communications Technology (ICOIACT) (pp. 186-191). IEEE.

19. Hu, Z., Hu, Y., & Li, H. (2025). Multi-Task Temporal Fusion Transformer for Joint Sales and Inventory Forecasting in Amazon E-Commerce Supply Chain. arXiv preprint arXiv:2512.00370.
20. Memon, Z., & Saini, I. (2024, December). A Comparative Survey of Blockchain-Based Security Mechanisms for OTA updates in CAVs. In 2024 IEEE/ACM 17th International Conference on Utility and Cloud Computing (UCC) (pp. 423-428). IEEE.
21. Gui, H., Wang, B., Lu, Y., & Fu, Y. (2025). Computational Modeling-Based Estimation of Residual Stress and Fatigue Life of Medical Welded Structures.
22. Merlin, P. W. (2020). Green Light for Green Flight: NASA's Contributions to Environmentally Friendly Aviation (No. HQ-E-DAA-TN76912). National Aeronautics and Space Administration.