



Article

Airport Surface Arrival and Departure Scheduling Using Extended First-Come, First-Served Scheduler

Bae-Seon Park ¹  and Hak-Tae Lee ^{2,*} 

¹ Industrial Science Technology Research Institute, Inha University, Incheon 21999, Republic of Korea; pearship@inha.ac.kr

² Department of Aerospace Engineering, Inha University, Incheon 21999, Republic of Korea

* Correspondence: haktae.lee@inha.ac.kr

Abstract: This paper demonstrates the effectiveness of the Extended First-Come, First-Served (EFCFS) scheduler for integrated arrival and departure scheduling by comparing the scheduling results with the recorded operational data at Incheon International Airport (ICN), Republic of Korea. The EFCFS scheduler can handle multiple capacity- or flow-rate-related constraints along the path of each flight, which is represented by a node-link graph structure, and can solve large-scale problems with low computational cost. However, few studies have attempted a systematic verification of the EFCFS scheduler by comparing the scheduling results with historical operational data. In this paper, flights are scheduled between gates and runways on the airport surface with detailed constraints such as runway wake turbulence separation minima and conflict-free taxiing. The scheduler is tested using historical flight data from 15 August 2022 at ICN. The input schedule is generated based on the flight plan data extracted from the Flight Operation Information System (FOIS) and airport surface detection equipment data, and the results are compared with the times extracted from the FOIS data. The scheduling results for 500 aircraft show that the average takeoff delay is reduced by about 19 min, while the average landing delay is increased by less than one minute when the gate occupancy constraint is not considered. The results also confirm that the EFCFS effectively utilizes the available time slots to reduce delays by switching the original departure or arrival orders for a small number of flights.



Citation: Park, B.-S.; Lee, H.-T. Airport Surface Arrival and Departure Scheduling Using Extended First-Come, First-Served Scheduler. *Aerospace* **2024**, *11*, 24. <https://doi.org/10.3390/aerospace11010024>

Academic Editors: Joost Ellerbroek and Michael Schultz

Received: 3 November 2023

Revised: 7 December 2023

Accepted: 24 December 2023

Published: 26 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: scheduling; FCFS; airport surface

1. Introduction

The first-come, first-served (FCFS) principle is a traditional approach to solving scheduling problems [1]. In general, an FCFS-based scheduler computes the target departure or arrival times for flights in the order in which they were initially requested and schedules each time as a single point [2,3]. For example, each flight in the queue is assigned a target takeoff time (TTOT) based on the runway separation constraints, and the target off-block time (TOBT) can be calculated by subtracting a predetermined nominal taxi time for the gate-runway combination. FCFS is simple in the sense that the scheduling results can be readily understood and can be considered equitable, as the departure or arrival order does not change.

Two major issues with the traditional FCFS scheduler are: first, that it is difficult to accommodate multi-point constraints along the flight path, and second, that it tends to miss available time slots in order to preserve the original order. Consequently, if the demand becomes higher, flights further down the queue can accumulate excessive delays.

Most of the recent research on scheduling algorithms are based on optimization, such as mixed-integer linear programming (MILP) [4–10]. Smeltink et al. [4] presented a MILP-based formulation to solve the airport taxi scheduling problem. Visser and Roling [5] investigated a similar MILP-based taxi scheduler with a multi-route option. In [6,7],

an optimization model for taxiway and runway scheduling was studied using a MILP approach, and Lee and Balakrishnan [8] proposed a MILP model to optimize both the taxiway and runway scheduling problems simultaneously. Bosson and Sun [9] suggested stochastic scheduling optimization considering uncertainty in the MILP formulation for surface routing and scheduling. Yang and Gao [10] also proposed a stochastic scheduling model based on MILP that can solve the ground movement problem (GMP) combining taxiway routing and gate allocation.

Schedulers using MILP-based optimizations usually provide optimal solutions, but they are computationally very costly and sometimes do not converge. So they are not suitable for large-scale problems or for operational situations for which a timely solution is required. To overcome these issues, researchers have been working in two main areas: one is improving the formulation [11–17], and the other is improving the efficiency of the solver [18–23].

Avella et al. [11] proposed time-indexed formulations that can be solved by a general-purpose mixed-integer programming (MIP)-based solver for the runway scheduling problem; they showed significant improvement in computation time. Prakash et al. [12] also improved computation performance of a MIP-based algorithm to solve aircraft sequencing problems using data-splitting that enables parallel computing. Wang et al. [14] introduced a chance-constrained programming-based optimization model combined with the quickest-path problem with time windows to solve the GMP while considering taxi time uncertainties. Zou et al. [15] and Ornek et al. [16] presented a way to formulate the problem in a pure-integer programming-based optimization model for taxi scheduling and the flight–gate assignment problem (FGAP). A multi-objective programming model was also investigated. Hu et al. [13] considered uncertainty theory for FGAP, and Dönmez et al. [17] used the results for terminal area management.

Ma et al. [18] adjusted the simulated annealing algorithm by combining time decomposition to efficiently solve large-scale problems. Yan et al. [19] improved a particle swarm optimization based on a receding horizon to solve the joint scheduling problem on the airport surface. Sun et al. [20] used a genetic algorithm to improve the computational speed for solving the bilevel programming-based optimization problem; their model simultaneously considers gate assignment, taxi path scheduling, and pushback time delay. Xu et al. [21] proposed a faster MIP solver that combined matrix approximation and an ant colony algorithm to solve aircraft arrival and departure scheduling problems. Zhang et al. [22] showed that adding cosine mutation and adaptive grouping to a tunicate swarm algorithm decreases the computation time when solving the gate assignment problem (GAP). Du et al. [23] solved the robust GAP to minimize ground conflict using a column-generation-based heuristic method for computational efficiency.

However, continuous efforts have been made to improve the FCFS scheduler. Meyn [24] was the first to propose a propagation-based multi-point scheduling methodology in a node–link structure; they established a framework to address two fundamental issues of the simple FCFS scheduler. In [25], the capabilities of their scheduler were demonstrated by imposing multi-point flow-rate constraints at multiple metering fixes along the flight path. Palopo et al. [26] introduced a capacity constraint represented by a maximum number of aircraft in a sector to study the traffic flow management problem in a dynamic airspace environment. Park (C) et al. [27] combined the work of Meyn [24] and Palopo et al. [26] so that the scheduler can handle both flow-rate-style constraints at nodes such as metering fixes and maximum-capacity constraints at links such as sector monitor alert parameters (MAPs) to develop an advanced FCFS scheduler, and they investigated the effectiveness of sector transit time control to efficiently utilize available time slots.

Park (B) et al. [28] proposed the extended FCFS (EFCFS) scheduler for surface operation; it can enforce minimum runway separation constraints based on the wake turbulence categories. In addition, in [28], the capability to explore multiple routes was added. It was demonstrated that delays can be further reduced by going around a congested junction or taxiway even though this increases the overall taxi distances. Park (B) [29] proposed a

change to the original propagation architecture of Meyn [24] to handle the link directionality constraint without using the negative aircraft count concept and to directly enforce the minimum along-trail separation between leading and trailing aircraft instead of using an indirect link capacity. In [29], iterative rescheduling, which takes advantage of lower computational cost, was investigated to demonstrate that the scheduler can be used in an actual operational environment. Phillips and Sadosky [30] enhanced [24] by enforcing a finite time interval to handle uncertainty and to be used for distributed scheduling.

Park (B) et al. [31] performed a study to compare the EFCFS with a MILP-based scheduler of Eun et al. [32]. First, each constraint was investigated to ensure the compatibility of the two schedulers, and then, a large number of scheduling problems were solved at Incheon International Airport (ICN). The results show that the schedules computed by EFCFS have, on average, about 18% larger delay. However, computation time was at least one order of magnitude smaller for the EFCFS.

In this paper, the EFCFS scheduler using the architecture of [29] is employed to validate the scheduling results with actual historical data. The scenarios were generated based on the actual schedules and aircraft types from the FOIS data and the taxi routes extracted from the airport surface detection equipment (ASDE) track data at ICN. Differences in takeoff times (TOTs) and landing times (LDTs) calculated by comparing the scheduling results with the initial plan information from FOIS were used as metrics. The results show that the average takeoff delay can be reduced by as much as about 19 min depending on the gate occupancy constraint. It is also confirmed that the scheduler effectively utilizes available time slots by switching the departure or landing order for a small number of flights.

The main contribution is that this work has established a systematic framework of utilizing the operational data (FOIS and ASDE data) to validate the performance of the EFCFS scheduler, whereas most of the previous work focused on developing the scheduling algorithm itself. The framework includes the comparison metrics, the generation of input schedules, and the construction of complicated constraints. In addition, the framework is demonstrated on a complex airport with a large number of flights.

After this introduction, Section 2 describes the EFCFS scheduler and its constraints on the nodes and links. Section 3 explains how the historical FOIS data are used to generate the reference schedule. Section 4 explain the scenario, and Section 5 shows the scheduling results. Section 6 concludes this paper.

2. Extended First-Come, First-Served Scheduler [28,29,31]

2.1. Baseline Algorithm

The EFCFS scheduler performs priority-based scheduling and provides the best solution for a given flight that satisfies all the constraints imposed by flights with higher priority. It is similar to the basic FCFS approach, but the most important feature of EFCFS is that the scheduler can generate solutions out of order rather than in the originally given order. This allows the scheduler to have performance similar to that of optimization-based schedulers.

Figure 1 shows the scheduling process of the EFCFS scheduler. The green and gray blocks represent the available and unavailable slots of the nodes and links from the results of fixing the schedules of the higher priority flights with constraints such as runway separation and along-trail spacing. Figure 1a shows the forward propagation process to determine the earliest possible arrival time at the destination node. All the available slots of the origin node A are propagated downstream along the path. The green areas in the links represent the feasible area computed by a forward propagation step. The forward feasible area has the shape of a trapezoid, where the slopes of the left and right boundaries represent the maximum and minimum transit speeds, respectively. This makes the available slots of the link exit node wider than the available slots of the link entry node, which allows flights to better utilize available slots to minimize the delay to the destination node. Some of the feasible areas are blocked by the unavailable slots of node B and link BC represented in gray. These unavailable slots are created for flights that were already scheduled before

the current flight. The earliest time of the available slots at the destination, node D, is the earliest possible arrival time of the flight, as shown in Figure 1a.

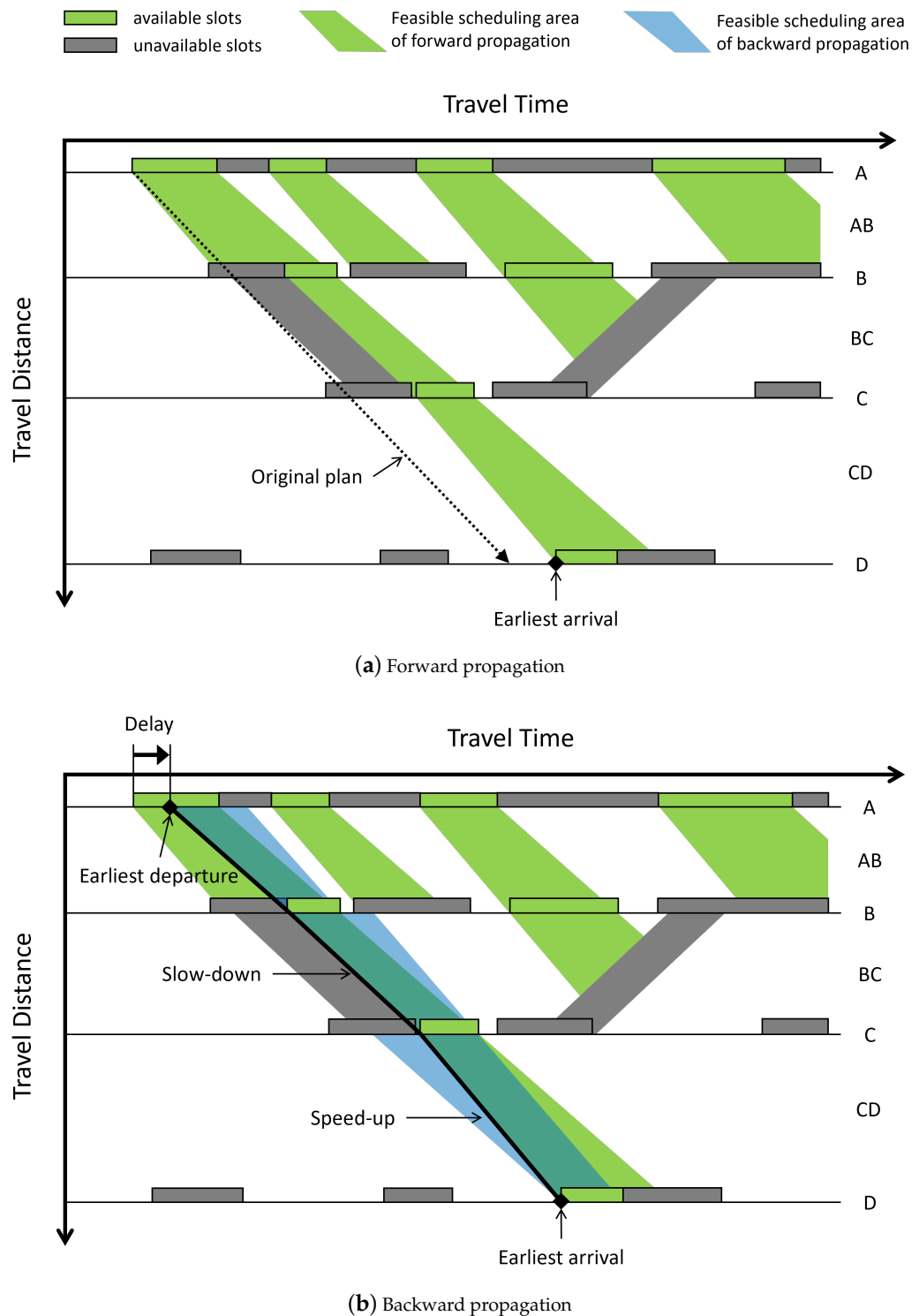


Figure 1. Propagation steps of the EFCFS scheduler for surface operations.

Once the earliest arrival time at node D is determined, the earliest possible departure time at the origin node can be computed by backward propagation, as shown in Figure 1b. The blue area represents the feasible scheduling area propagated backward from the destination node D to the origin node A. The backward-feasible area is the reverse of the

forward-feasible area in the sense that the left and right slopes are swapped. Flights can depart early and transit slow or depart late and transit fast to meet the available time slots.

The final feasible region is the intersection of the forward- and backward-feasible areas. The left boundary of the overlapping areas is the scheduling solution from the origin to the destination, and the first entry time of the schedule is the earliest departure time. The unimpeded entry time at each node is calculated from the original planned departure time and the nominal taxi speed. The difference between the unimpeded entry time and the scheduling solution is the delay time at each node.

The mathematical formulation of the EFCFS algorithm is described in the Appendix A.

2.2. Scheduling Constraints

In this section, constraints to achieve proper runway separation and conflict-free taxiing are explained. Two fundamental constraints are the passing rate constraint at a node and the minimum along-trail separation at a link.

2.2.1. Node Constraint

The node constraint can be defined as a passing rate: the number of aircraft that transit the node per unit time. The passing rate of the node can be converted to the minimum time interval between two consecutive aircraft that pass through the same node. In this study, this time interval is referred to as the blocking time.

Figure 2 shows that the blocking time t_b is applied to the scheduled time t . The node is unavailable for the interval $[t - t_b, t + t_b]$.

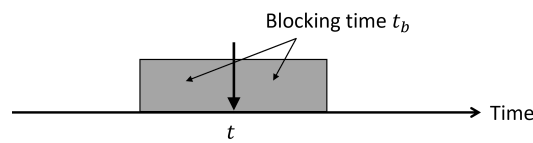


Figure 2. Blocking time constraint at the node.

The runway node requires additional constraints. Runway occupancy time (ROT) is the time it takes a flight to takeoff or land. Once a flight’s takeoff or landing schedule is fixed, ROT is added to the calculated entry time at each runway node along the runway to make the runway unavailable for ROT, as shown in Figure 3. ROT is applied to all runway nodes in the takeoff or landing path by adding it after the determined entry time, as shown in Figure 3.

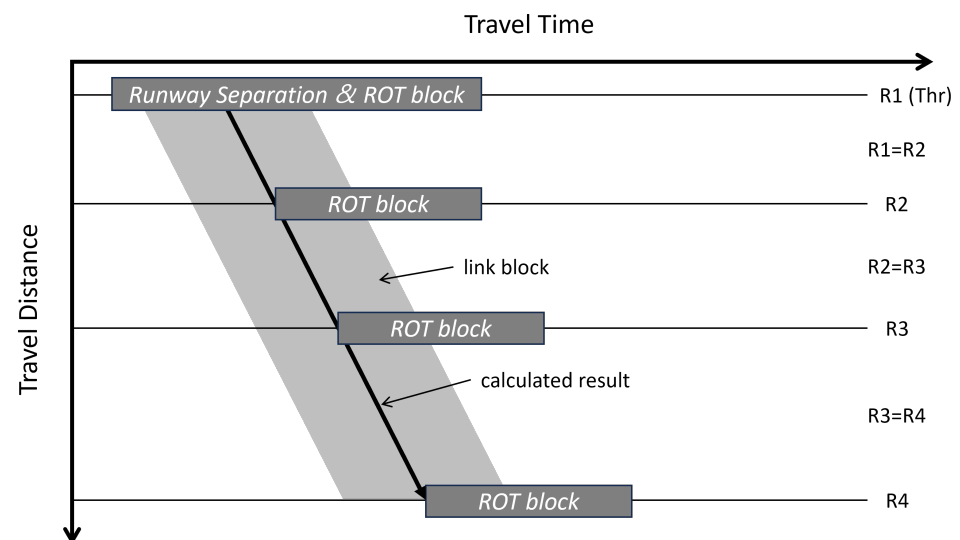


Figure 3. Runway constraints for the runway nodes.

The runway threshold node also considers the runway separation minima [28]. Figure 4 explains how these constraints can be applied. In Figure 4a, the heavy class aircraft, which was scheduled before the current flight due to higher priority, has an assigned arrival time of t_H . If the current aircraft arrives before t_H , the size of the unavailable slot is Δt_{LH} , where the leading aircraft is light and trailing aircraft is heavy. On the contrary, if the current aircraft arrives after t_H , the size of the unavailable slot is Δt_{HL} , where the leading aircraft is heavy and trailing aircraft is light. Generally, Δt_{HL} is larger than Δt_{LH} . So the threshold is not available for the interval $[t - t_{LH}, t + t_{HL}]$. Figure 4b shows a more complex situation, where a medium class aircraft is being scheduled after two other aircraft were assigned at times t_H and t_L . In this case, unavailable slots are generated for each aircraft, and the union of all the slots becomes unavailable.

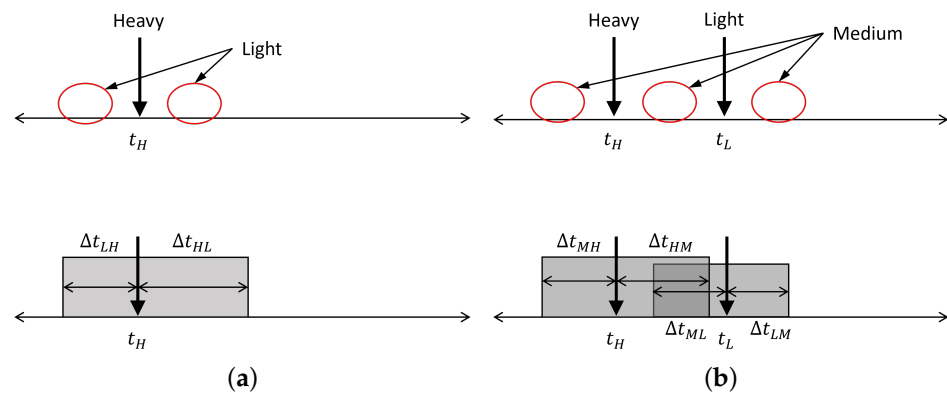


Figure 4. Applying runway separation at the threshold. (a) Light class is being scheduled after heavy class was scheduled. (b) Medium class is being scheduled after the heavy and light classes were scheduled.

2.2.2. Link Constraint

Once a flight’s schedule is fixed, it is represented by a line from the upper left to the lower right, as represented by a solid back line in Figure 1b. Consequently, along-trail spacing inside a link can be enforced by blocking a time interval, t_{lb} , before and after the solid line to form an area in the shape of a parallelogram, as shown in Figure 5. This architecture proposed in [29] ensures that once an aircraft enters a link, no other aircraft can overtake or enter from the opposite direction until it exits the link. If a parallelogram area is from the upper right to the lower left as in the third one in Figure 5, it means this flight traversed the link in the opposite direction.

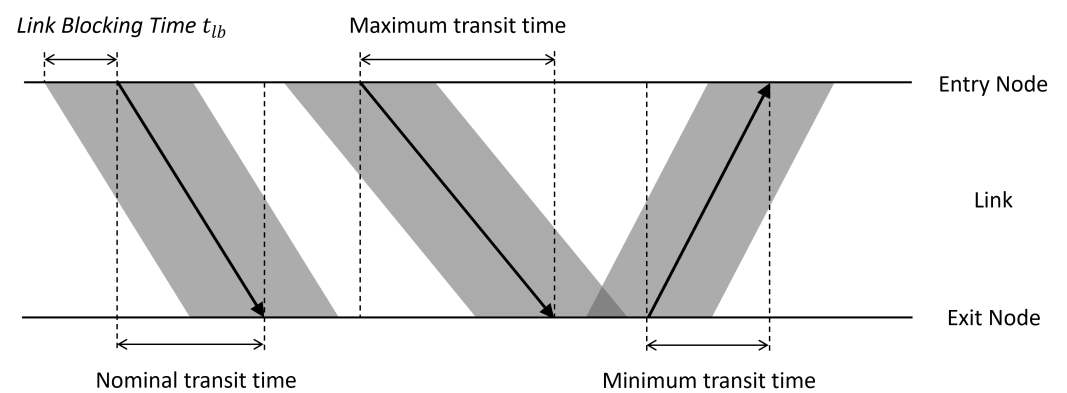


Figure 5. Link transit blocks.

3. Schedule Generation

To generate realistic scenarios, this paper uses FOIS data [33], which are typical data that provide flight schedules, and ASDE data. FOIS provides the basic flight information,

such as the callsign, aircraft type, origin airport and gate, and destination airport and gate. However, FOIS does not contain runway information. FOIS is divided into two categories: departure and arrival. The departure FOIS contains the estimated time of departure (ETD) and actual time of departure (ATD) at the origin airport and the estimated time of arrival (ETA) at the destination airport. Comparing the FOIS times with the ASDE tracks confirms that ETD and ATD are the times at the departure runway. The arrival FOIS provides the ETA and the actual time of arrival (ATA) at the destination airport and the ETD at the origin airport. Similar to the departure FOIS, ETA and ATA are scheduled at the arrival runway.

In addition to crosschecking the FOIS times, the ASDE data are used to extract the actual routes on the surface. ASDE includes dense track data of the flights and all other objects that moved on the surface. The surface route of each flight is extracted by comparing the track positions with the surface node–link model.

Among the 500 flights used in this study, seven flights in FOIS do not have usable track data. For those flights, the most commonly used departure or arrival runways are assigned and the minimum distance route between the runway and the gate is used as a substitute.

The original planned flight schedules are generated by combining flight information, ETA, ETD, and extracted routes. Since the ASDE track data are the result of traffic control and scheduling, ATA, ATD, and tracking times are not appropriate for validating the performance of the presented scheduler. Consequently, in this paper, the ETDs and ETAs were used to generate the original plan. For departure, the ETD is the planned runway takeoff time, STOT, so the scheduled OBT (SOBT) is calculated by subtracting the scheduled taxi-out time (SXOT) from the ETD. For arrivals, the ETA is the planned landing time, SLDT, at the runway. The scheduled IBT (SIBT) for arrival is the sum of SLDT and the scheduled taxi-in time (SXIT). Scheduled taxi times—SXOT and SXIT—are the sum of all nominal link transit times in the route from the runway threshold to the gate. The arrival route includes runway links from touchdown to the runway exit; however, the departure route ends at the runway threshold and does not contain any runway links. The nominal transit time of each surface link is calculated by dividing the length of the link by the nominal taxi speeds, which are set to 5, 10, and 15 knots for the gate, ramp, and taxiway links, respectively. A speed of 150 knots is assumed for the runway links.

Table 1 is a sample plan generated using the departure track in Figure 6a that shows the arrival route of Flight1 from runway 16R to gate G21. Figure 6b shows the departure route of Flight2 from gate G14 to the takeoff from runway 16L. The nominal transit times of each link are the differences between the entry times and exit times.

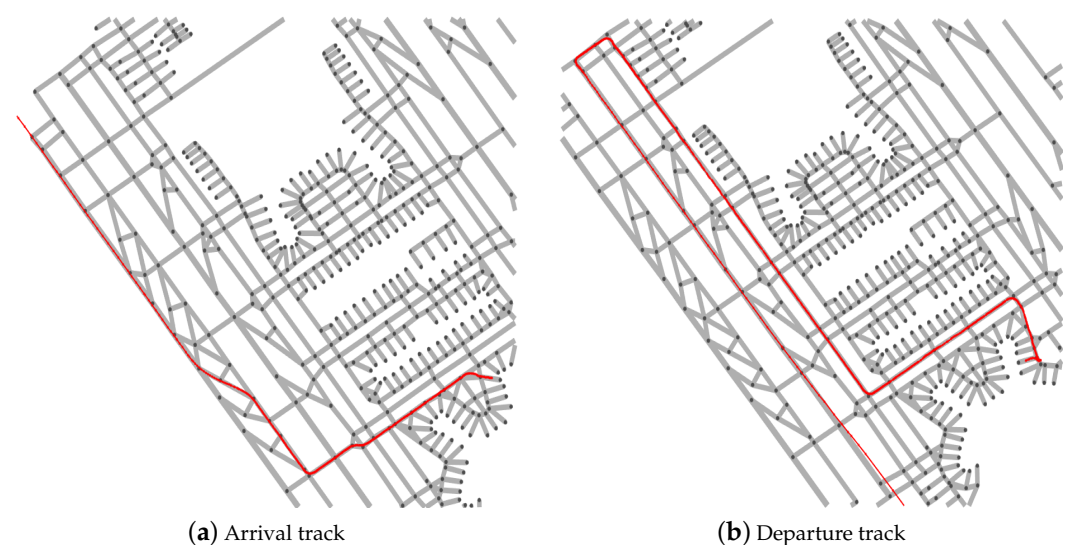


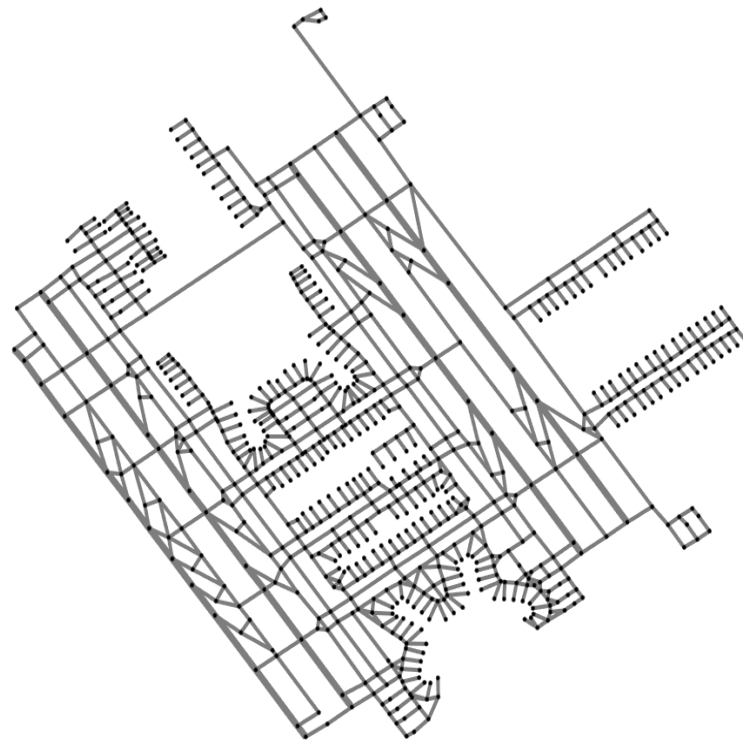
Figure 6. ASDE tracks on the surface.

Table 1. A sample schedule of tracks in Figure 6. The unit of time is unix time in seconds.

| Flight | Entry Node | Exit Node | Transit Link | Entry Time | Exit Time |
|---------|------------|-----------|--------------|------------|------------|
| Flight1 | 16R | RWY27 | RWY27–16R | 1660544580 | 1660544582 |
| Flight1 | RWY27 | RWY28 | RWY27–RWY28 | 1660544582 | 1660544585 |
| Flight1 | RWY28 | RWY29 | RWY28–RWY29 | 1660544585 | 1660544589 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Flight1 | RWY32 | TX140 | RWY32–TX140 | 1660544608 | 1660544631 |
| Flight1 | TX140 | TX141 | TX140–TX141 | 1660544631 | 1660544666 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Flight1 | RP313 | RP41 | RP41–RP313 | 1660544935 | 1660544963 |
| Flight1 | RP41 | RP31 | RP31–RP41 | 1660544963 | 1660544991 |
| Flight1 | RP31 | G21 | G21–RP31 | 1660544991 | 1660545046 |

4. Problem Setup

To evaluate the EFCFS scheduler, surface scheduling for departures and arrivals was tested at ICN. Figure 7 shows the ICN node–link model, which has 1006 links and 775 nodes with 262 gates and 8 runway thresholds for 2 pairs of parallel runways. The node–link model was created manually using the aeronautical information publication document and satellite images from Google Earth.

**Figure 7.** ICN node–link model.

The originally planned schedule was generated using FOIS and ASDE data at ICN on 15 August 2022. The schedule includes 250 departures and 250 arrivals, with 150 medium, 343 heavy, and 7 super-heavy flights. Table 2 shows the total fleet mix of the generated schedule.

Tables 3 and 4 show the node constraints used for EFCFS computation. No additional occupancy constraints are applied to the gate nodes because the SIBTs and SOBTs of the input schedules already reflect the gate occupancy, and the scheduler does not allow TOBT earlier than SOBT. The ramp and taxi nodes are constrained by a ten-second blocking time. The runway nodes are constrained by a 30-second blocking time and runway occupancy times.

Table 2. Fleet mix.

| | Runways | Medium | Heavy | Super | Total |
|------------|---------|--------|-------|-------|-------|
| Departures | 16L | 51 | 111 | 3 | 165 |
| | 34R | 23 | 45 | | 68 |
| | 33R | 1 | 15 | 1 | 17 |
| Arrivals | 16R | 45 | 57 | 1 | 103 |
| | 34L | 25 | 38 | 1 | 64 |
| | 15L | 4 | 38 | 1 | 43 |
| | 33R | 1 | 39 | | 40 |
| Total | | 150 | 343 | 7 | 500 |

Table 3. Node blocking times in seconds.

| Types | Blocking Times |
|--------|----------------|
| Gate | 0 |
| Ramp | 10 |
| Taxi | 10 |
| Runway | 30 |

Table 4. Runway occupancy times in seconds.

| | Departure | Arrival |
|--------|-----------|---------|
| Light | 85 | 80 |
| Medium | 60 | 50 |
| Heavy | 50 | 45 |
| Super | 50 | 45 |

The runway separation criteria are considered as additional constraints for the runway threshold nodes. In this paper, modified runway separations based on the separation standard were used [34]. Table 5 shows the runway separations for single runways that are applied to the scheduler. The separations between departure after arrival and arrival after departure are set to one minute. The separations between departures and between arrivals on adjacent runways are set to two minutes. For the opposite direction, a two-minute separation is applied between departures and between arrivals for both the single runway and adjacent runway operations. All other cases are set to zero separation. Table 6 shows the total runway separations for ICN that were applied to the scheduler.

Table 5. Runway separations on a single runway and in the same direction. (a) Departure after departure, (b) Arrival after arrival.

| (a) | | | | |
|------------|-------|--------|-------|-------|
| Lead/Trail | Light | Medium | Heavy | Super |
| Light | 120 | 120 | 120 | 120 |
| Medium | 120 | 120 | 120 | 120 |
| Heavy | 180 | 120 | 120 | 120 |
| Super | 180 | 180 | 120 | 120 |
| (b) | | | | |
| Lead/Trail | Light | Medium | Heavy | Super |
| Light | 120 | 120 | 120 | 120 |
| Medium | 180 | 120 | 120 | 120 |
| Heavy | 180 | 120 | 120 | 120 |
| Super | 180 | 180 | 120 | 120 |

Table 6. Total runway separations in seconds.

| Lead/Trail | Departures | | | | | | | | Arrivals | | | | | | | | |
|------------|------------|-------|-----|-----|-----|-----|-----|-----|----------|--------|-----|-----|-----|-----|-----|-----|-----|
| | 33L | 33R | 34L | 34R | 15L | 15R | 16L | 16R | 33L | 33R | 34L | 34R | 15L | 15R | 16L | 16R | |
| Departures | 33L | (a) * | 120 | - | - | 120 | 120 | - | - | 60 | 0 | - | - | 120 | 0 | - | - |
| | 33R | 120 | (a) | - | - | 120 | 120 | - | - | 0 | 60 | - | - | 0 | 120 | - | - |
| | 34L | - | - | (a) | 120 | - | - | 120 | 120 | - | - | 60 | 0 | - | - | 120 | 0 |
| | 34R | - | - | 120 | (a) | - | - | 120 | 120 | - | - | 0 | 60 | - | - | 0 | 120 |
| | 15L | 120 | 120 | - | - | (a) | 120 | - | - | 120 | 0 | - | - | 60 | 0 | - | - |
| | 15R | 120 | 120 | - | - | 120 | (a) | - | - | 0 | 120 | - | - | 0 | 60 | - | - |
| | 16L | - | - | 120 | 120 | - | - | (a) | 120 | - | - | 120 | 0 | - | - | 60 | 0 |
| | 16R | - | - | 120 | 120 | - | - | 120 | (a) | - | - | 0 | 120 | - | - | 0 | 60 |
| Arrivals | 33L | 60 | 0 | - | - | 120 | 0 | - | - | (b) ** | 120 | - | - | 120 | 120 | - | - |
| | 33R | 0 | 60 | - | - | 0 | 120 | - | - | 120 | (b) | - | - | 120 | 120 | - | - |
| | 34L | - | - | 60 | 0 | - | - | 120 | 0 | - | - | (b) | 120 | - | - | 120 | 120 |
| | 34R | - | - | 0 | 60 | - | - | 0 | 120 | - | - | 120 | (b) | - | - | 120 | 120 |
| | 15L | 120 | 0 | - | - | 60 | 0 | - | - | 120 | 120 | - | - | (b) | 120 | - | - |
| | 15R | 0 | 120 | - | - | 0 | 60 | - | - | 120 | 120 | - | - | 120 | (b) | - | - |
| | 16L | - | - | 120 | 0 | - | - | 60 | 0 | - | - | 120 | 120 | - | - | (b) | 120 |
| | 16R | - | - | 0 | 120 | - | - | 0 | 60 | - | - | 120 | 120 | - | - | 120 | (b) |

* Represents the values in Table 5a. ** Represents the values in Table 5b.

To ensure the runway separation constraints are aligned with the actual operation, historical runway separations are investigated using the ASDE data. Figure 8 shows the distributions of the actual runway separations between departures and between arrivals on runways 15R/33L and 15L/33R from the one-month ASDE data, which confirms that the constraints given in Table 5 are reasonable.

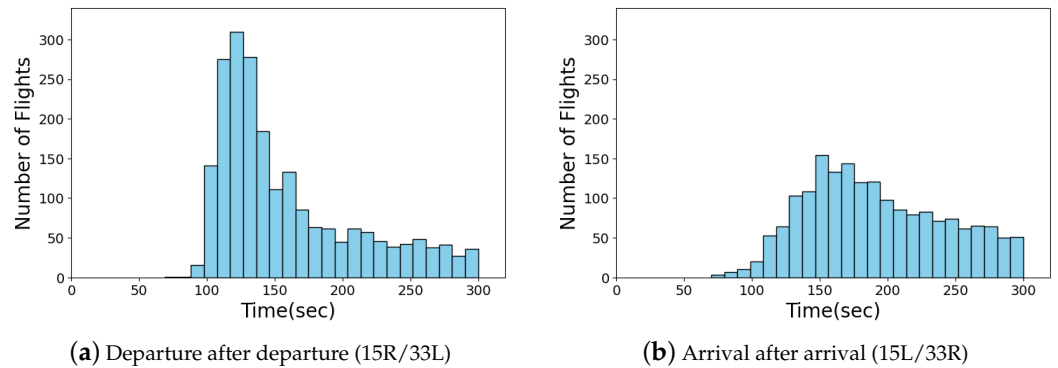


Figure 8. Distributions of the separations on runways 15R/33L and 15L/33R.

The link blocking time is set to ten seconds for all types of links to enforce a minimum ten-second along-trail separation between the taxiing aircraft. The taxi speed can be reduced by up to 10%, except on the runway, which increases the link transit time by up to 10% of the nominal transit time.

5. Scheduling Results

The EFCFS scheduler provides different results depending on the priorities. Among several prioritization strategies, in this study, flights are scheduled based on the nominal priority, which is the order of the planned SOBT and SLDT.

Delays are measured by computing the difference between the scheduled time and the target time, which is calculated by the scheduler. Figure 9 shows the distributions of four delays—DOBT, DIBT, DTOT, and DLDT—which are defined in Equations (1)–(4). Table 7 summarizes the average and maximum delay times. Most delays are between zero and one minute, and the average DOBT, DIBT, DTOT, and DLDT are 0.79, 0.18, 1.01, and 0.17 min, respectively. The maximum DOBT, DIBT, DTOT, and DLDT are 12.43, 6.9, 13, and 6.9 min,

respectively. The average DTOT and DIBT are larger than the average DOBT and DLDT. This shows that the scheduler reduced the taxi speed of the flights to better utilize available time slots. Mostly, the departure flights are adjusted while the arrivals are not significantly affected by the scheduler.

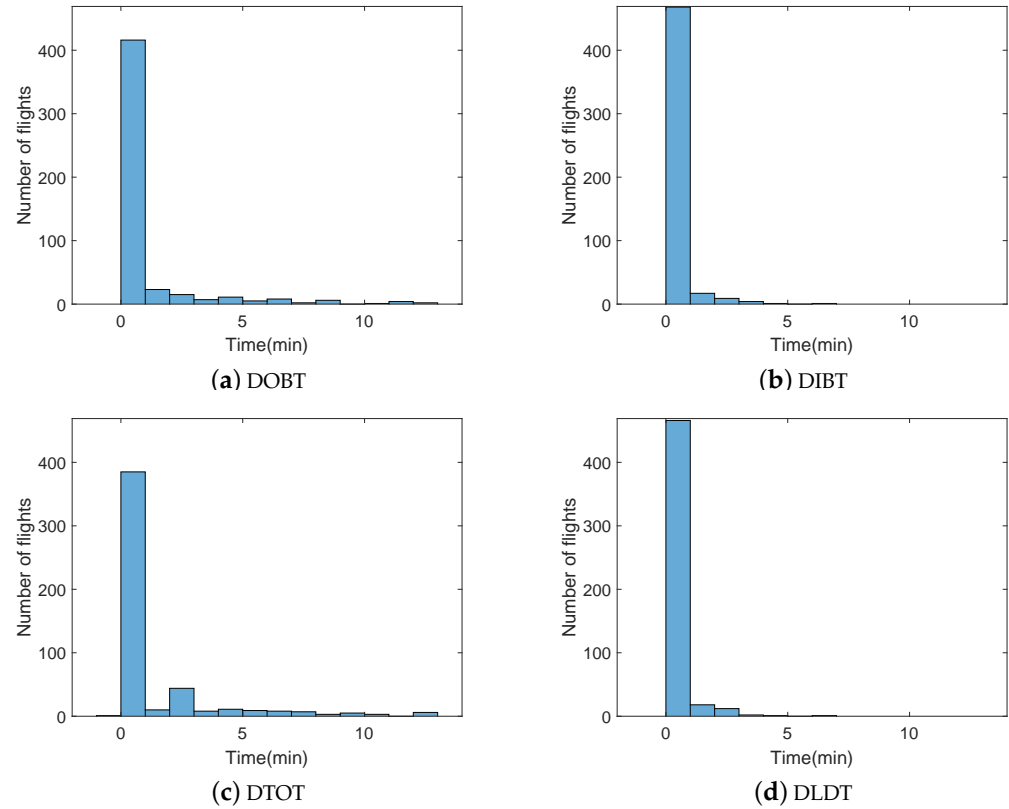


Figure 9. Delay distributions.

Table 7. Delays in minutes.

| | DOBT | DTOT | DIBT | DLDT |
|---------|-------|------|------|------|
| Average | 0.79 | 1.01 | 0.18 | 0.17 |
| Maximum | 12.43 | 13 | 6.9 | 6.9 |

Since the speed of 150 knots is used for all runway links, the runway occupancy time will be slightly smaller, especially for arrivals. The runway link speeds are reduced to 100 and 50 knots to test the impact, and the results do not show any significant differences. Therefore, the original value of 150 knots is used for all subsequent runs.

$$DOBT = TOBT - SOBT \quad (1)$$

$$DTOT = TTOT - STOT \quad (2)$$

$$DIBT = TIBT - SIBT \quad (3)$$

$$DLDT = TLDT - SLDT \quad (4)$$

Figure 10 shows the runway throughputs for the scheduling results. Figure 10a shows the departure runway throughputs, and Figure 10b shows the arrival runway throughputs. The maximum throughput of each runway is 30 flights per hour, and there is no runway separation constraint violations in the scheduling results. Similarly, no other constraint violation is detected in the scheduler output.

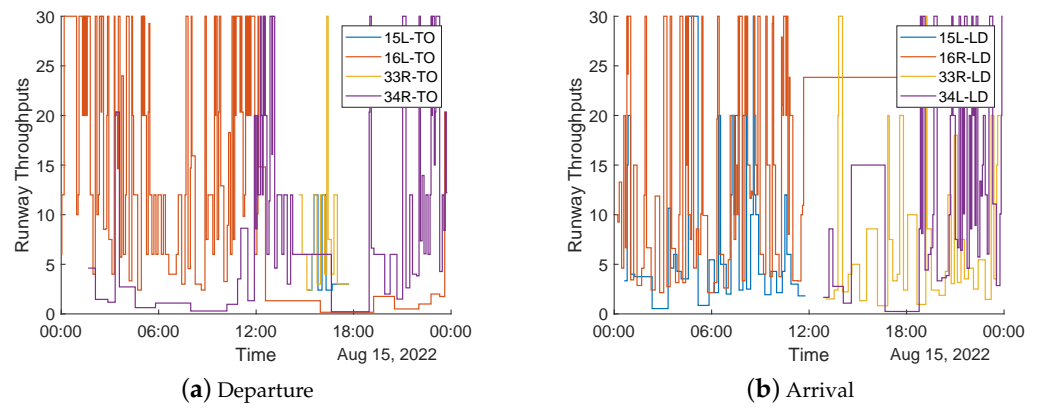


Figure 10. Runway throughputs.

Figure 11a shows a portion of the STOTs and TTOTs. Solid lines denote zero delay, and dashed line denotes flights with a TOT adjusted by the scheduler. The square markers show that the flights with the same planned time have been adjusted for proper spacing. Flights marked with a red x in the TTOT column represent an order change. It can be observed that the scheduler took advantage of the available time slots by changing the order of several flights. Similar results can be observed for landing flights, as shown in Figure 11b.

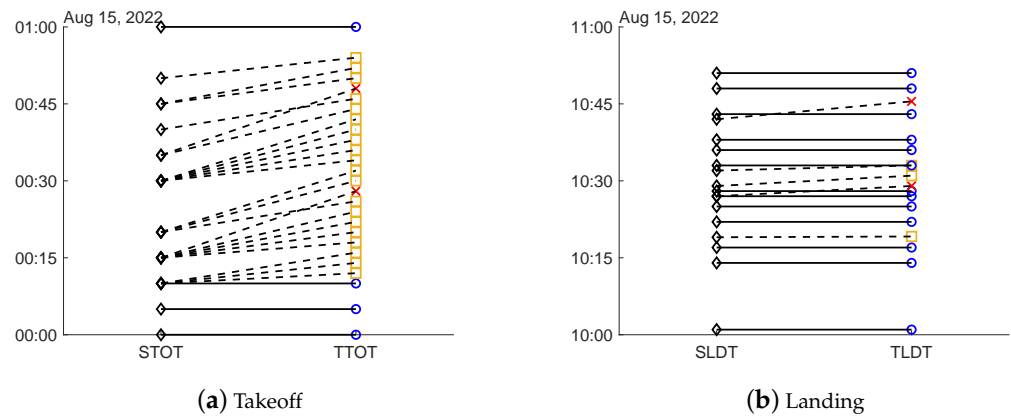


Figure 11. Change to departure and arrival orders.

5.1. Comparison with the Operational Data

The scheduling results are compared to the actual FOIS data to validate performance of the scheduler. ATD and ATA are equal to the actual TOT (ATOT) and LDT (ALDT), respectively. To compare the results with the scheduler output, the difference between ATOT and TTOT is used for departures, and the difference between ALDT and TLDT is used for arrivals. If the difference is positive, it can be judged that the scheduler provides a better time than the actual data.

Figure 12 show the time difference distributions between the scheduling result and the actual data. In Figure 12a, the differences in TOT are skewed to the positive side. On the contrary, the majority of the LDT differences are mostly zero and are located on the slightly negative side in Figure 12b. Table 8 shows the average and maximum differences. The average differences for TOT and LDT are 18.76 and -0.65 min, respectively. The maximum differences for TOT and LDT are 103.85 and 41 min, and the minimum differences for OBOT and LDT are -10 and -28 min, respectively. For this particular example, the EFCFS significantly reduced the departure delays while having negligible impact on the landing delays.

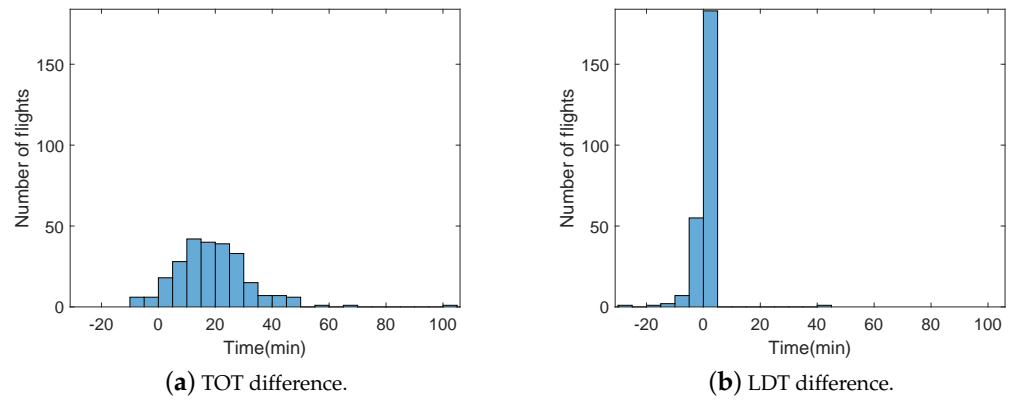


Figure 12. Difference distributions of TOT and LDT between scheduling results and the actual data.

Table 8. Differences between scheduler and actual data in minutes. Positive values mean that the scheduler provided better results.

| | Takeoff Time Differences | Landing Time Differences |
|---------|--------------------------|--------------------------|
| Average | 18.76 | −0.65 |
| Maximum | 103.85 | 41 |
| Minimum | −10 | −28 |

5.2. Gate Occupancy Implementation

The comparison results without considering gate occupancy show that the EFCFS scheduler is effective at reducing departure delays. Since the delay could have been reduced by the unconstrained gate nodes, gate occupancy times are investigated for the flights that arrive and depart ICN on the same day. Among 500 flights, 143 flights are identified. Figure 13 shows the distributions of the intervals between ATA and ATD of those flights. As these time intervals are between landing and takeoff, actual gate occupancy times are smaller. It can be observed that for the most of the flights, the time interval is between 1.5 and 2.5 h, which is consistent with the typical gate occupancy times for international flights.

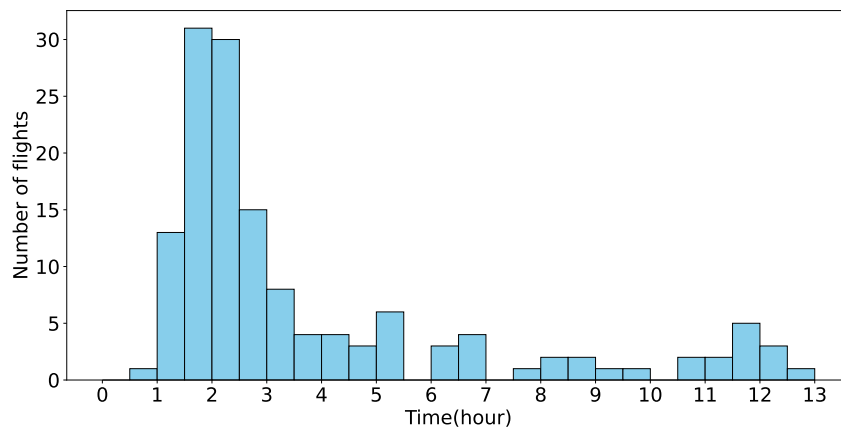


Figure 13. Distribution of the intervals between ATA and ATD of connecting flights.

To study the impact of gate occupancy constraints, gates are blocked for t_{goc} starting from the TIBT for inbound flights, as shown in Figure 14, and all the flights are rescheduled. The values for t_{goc} is set to one or two hours based on the trend shown in Figure 13, and the outcomes are compared to the previous results without the gate occupancy constraints in order to evaluate the impacts.

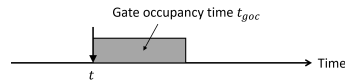


Figure 14. Gate occupancy after the gate in-block.

Table 9 shows the comparison of the scheduling results by the gate constraints. There was no drastic change for arrivals, but departure delays increase as the gate occupancy time increases. For departures, the averages for DOBT and DTOT increase from 0.79 and 1.01 to 2.41 and 2.64 min, respectively, when the gate is occupied by arrivals for one hour. And the average DOBT and DTOT significantly increase to 8.89 and 9.11 min when the gate occupancy time increases to two hours. The standard deviations also increase as the gate occupancy time increases, similar to the average delays.

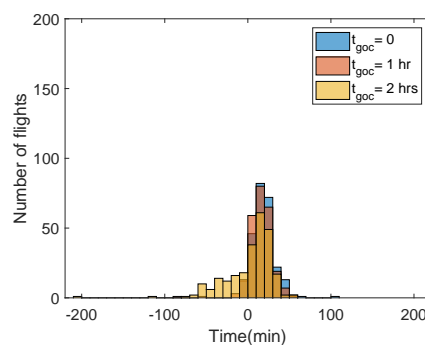
Table 9. Comparison of the scheduling results (in minutes).

| | | Unconstrained | 1-h Occupancy | 2-h Occupancy |
|------|---------------------|---------------|---------------|---------------|
| DOBT | Averages | 0.79 | 2.41 | 8.89 |
| | Standard deviations | 2.08 | 7.5 | 22.2 |
| DTOT | Averages | 1.01 | 2.64 | 9.11 |
| | Standard deviations | 2.35 | 7.59 | 22.2 |
| DIBT | Averages | 0.18 | 0.41 | 1.07 |
| | Standard deviations | 0.65 | 3.75 | 8.26 |
| DLDT | Averages | 0.17 | 0.4 | 1.05 |
| | Standard deviations | 0.63 | 3.71 | 8.19 |

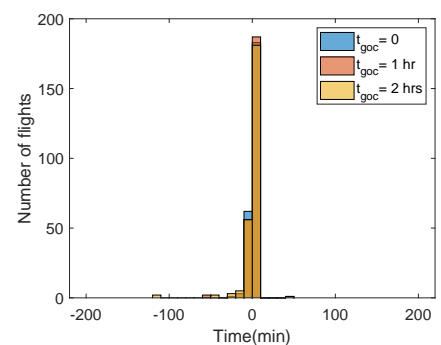
Table 10 summarizes the comparisons of the scheduling results and the actual data for the gate constraints. The scheduler produced results similar to the actual data as the gate occupancy time increased. Figure 15 shows the takeoff and landing time difference distributions between the scheduler and the actual data for the gate constraints. The distributions slide from positive to negative and also become wider as the occupancy time increases.

Table 10. Summary of the scheduler evaluations (in minutes). Positive values mean that the scheduler provided better results.

| | | Unconstrained | 1-h Occupancy | 2-h Occupancy |
|-------------------------|---------|---------------|---------------|---------------|
| Takeoff time difference | Average | 18.76 | 15.51 | 2.55 |
| | Maximum | 103.85 | 56.73 | 56 |
| | Minimum | −10 | −80.43 | −200.43 |
| Landing time difference | Average | −0.65 | −1.11 | −2.4 |
| | Maximum | 41 | 41 | 41 |
| | Minimum | −28 | −59.55 | −119.55 |



(a) TOT differences.



(b) LDT differences.

Figure 15. Comparison of difference distributions.

6. Conclusions

In this paper, the performance of the EFCFS scheduler is demonstrated using a realistic scenario at ICN based on historical data. The originally input schedule as well as various constraints are carefully constructed to reflect the current operational limits. Comparisons with operational data show that the scheduler effectively utilizes available time slots by switching the arrival or departure orders of a small number of flights and by making small adjustments to the transit speeds to significantly reduce delays.

As the EFCFS scheduler is significantly faster than any optimization-based scheduler, especially when the problem size is large, it can be a useful tool in real-world operations. With the validation framework established in this work, further studies will be conducted in order to make improvements in several areas, including refining the constraints based on an analysis of historical data, extending the domain up to the en-route merge points to reflect various metering constraints in the terminal area, and integrating route-finding so that multiple taxi routes can be explored to further reduce delays.

Author Contributions: Conceptualization, B.-S.P. and H.-T.L.; methodology, B.-S.P. and H.-T.L.; data curation, B.-S.P.; formal analysis, B.-S.P. and H.-T.L.; software, B.-S.P.; supervision, H.-T.L.; visualization, B.-S.P.; writing—original draft preparation, B.-S.P.; writing—review and editing, H.-T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by a Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (grant Nos. RS-2020-KA158275 and RS-2021-KA163373).

Data Availability Statement: The data is contained within this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Mathematical Formulations of the EFCFS Algorithm

The input variables of the EFCFS scheduler are the node-link model, which is the set of all nodes and links, and the flight plan, which is the set of all flights that have an assigned route. The input variables can be represented as

$$\mathbf{N} = \{N_i\}, 1 \leq i \leq m_N \quad (\text{A1})$$

$$\mathbf{L} = \{L_j\}, 1 \leq j \leq m_L \quad (\text{A2})$$

$$\mathbf{F} = \{F^k\}, 1 \leq k \leq m_F \quad (\text{A3})$$

$$\mathbf{R} = \{\mathbf{R}^k\}, 1 \leq k \leq m_F \quad (\text{A4})$$

where \mathbf{N} is the set of all nodes, \mathbf{L} is the set of all links, \mathbf{F} is the set of all flights, \mathbf{R} is the set of all routes, and \mathbf{R}^k is the route of flight F^k . \mathbf{F} is sorted by the scheduling priority of the flight such that a smaller k means higher priority. The variable m_N is the total number of nodes, m_L is the total number of links, and m_F is the total number of flights.

Each route of flight F^k is a sequence of nodes, and it can be represented as

$$\begin{aligned} \mathbf{R}^k &= (N_1^k, N_2^k, \dots, N_{n_k+1}^k), N_i^k \in \mathbf{N} \\ &= (L_1^k, L_2^k, \dots, L_{n_k}^k), L_j^k \in \mathbf{L} \end{aligned} \quad (\text{A5})$$

where N_i^k and L_j^k are the i -th node and j -th link, respectively, in route \mathbf{R}^k . The variable n_k is the number of links in route \mathbf{R}^k , so the number of nodes in route \mathbf{R}^k is $n_k + 1$.

In Equation (A5), L_i^k is the link between N_i^k and N_{i+1}^k . The sequence of the entry time and exit time, which is a 'transit' through link L_j^k , can be represented as

$$\mathbf{t}_j^k = (t_e^k, t_x^k) \quad (\text{A6})$$

where t_e^k is the entry time of flight F^k at link L_j^k , and t_x^k is the exit time of flight F^k at link L_j^k .

The node and link constraints are defined as follows:

$$\mathbf{CCN}_i^k = \bigcup_{l=1}^k \mathbf{CN}_i^l \tag{A7}$$

$$\mathbf{CCL}_j^k = \bigcup_{l=1}^k \mathbf{CL}_j^l \tag{A8}$$

where \mathbf{CCN}_i^k and \mathbf{CCL}_j^k are cumulative constraints at node N_i^k and link L_j^k , respectively, and \mathbf{CN}_i^k and \mathbf{CL}_j^k are unavailable time slots at node N_i^k and link L_j^k , respectively, imposed by reserving time slots for flight F^k . Figure A1 shows the node and link constraints defined in Equations (A7) and (A8). For runways, node constraints are determined considering the runway occupancy time and the wake-turbulence-category-based separation requirements at the threshold, as shown in Figures 3 and 4.

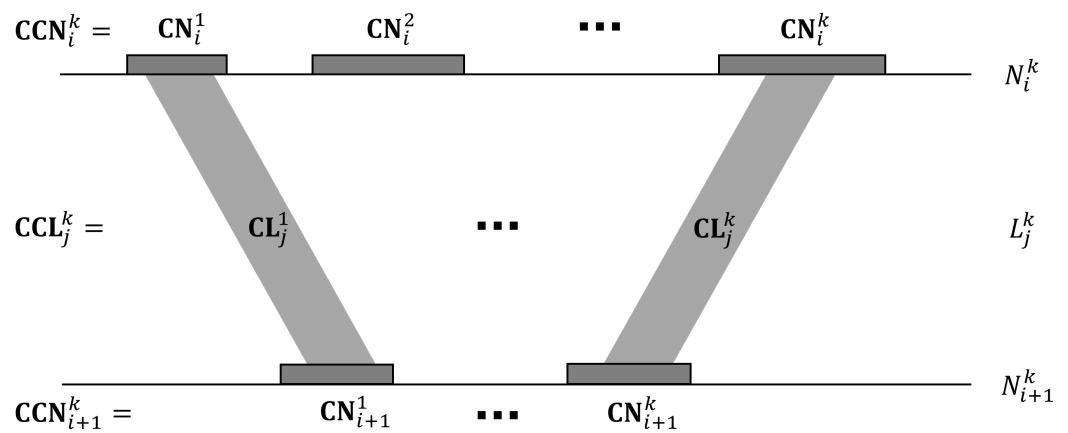


Figure A1. Cumulative node and link constraints in Equations (A7) and (A8).

Appendix A.1. Forward Propagation

For flight F^k , the objective of forward propagation is to find all transits through the given route that minimize the arrival time at the end node:

Find

$$\mathbf{t}_{forward}^k = (\mathbf{t}_1^k, \mathbf{t}_2^k, \dots, \mathbf{t}_{n_k}^k) \tag{A9}$$

Minimizes

$$t_{x_{n_k}}^k \tag{A10}$$

Subject to

$$t_{e_1}^k \geq SOBT^k \tag{A11}$$

$$t_{x_j}^k = t_{ef_{j+1}}^k \tag{A12}$$

$$\mathbf{t}_i^k \cap \mathbf{CCN}_i^{k-1} = \emptyset \tag{A13}$$

$$\mathbf{t}_j^k \cap \mathbf{CCL}_j^{k-1} = \emptyset \tag{A14}$$

$$t_e^k + \alpha t_n^k \leq t_x^k \leq t_e^k + \beta t_n^k \quad (0 \leq \alpha \leq 1, \beta \geq 1) \tag{A15}$$

where \mathbf{t}^k is the set of all transits through the route \mathbf{R}^k , $t_{x_{n_k}}^k$ is the exit time at the last link $L_{n_k}^k$ in route \mathbf{R}^k , t_{n_j} is the nominal transit time of link L_j , α is the speed-up factor, and β is the slow-down factor.

Equations (A11)–(A15) are the constraints for forward propagation. Equation (A11) constrains flight F_k so that it cannot depart before its SOBT. Equation (A12) is the continuity of transit, which represents that the exit time of link L_j^k is equal to the entry time of the next link, L_{j+1}^k . Equations (A13) and (A14) enforce the node and link constraints. Equation (A15) is the constraint to enforce the flight to adjust transit speeds within the given range.

Appendix A.2. Backward Propagation

The objective of backward propagation is to find the earliest departure time at the start node that satisfies all the constraints using the solution of forward propagation.

Find

$$\mathbf{t}_{backward}^k = (\mathbf{t}_1^k, \mathbf{t}_2^k, \dots, \mathbf{t}_{n_k}^k) \quad (\text{A16})$$

Minimizes

$$t_e^{k_1} \quad (\text{A17})$$

Subject to Equations (A12)–(A14) and,

$$t_{x_{n_k}}^k = (t_{x_{n_k}}^k)_{forward} \quad (\text{A18})$$

$$t_{x_j}^k - \beta t_{n_j}^k \leq t_e^{k_j} \leq t_{x_j}^k - \alpha t_{n_j}^k \quad (0 \leq \alpha \leq 1, \beta \geq 1) \quad (\text{A19})$$

Equations (A18) and (A19) are the constraints for backward propagation. Equation (A18) represents the earliest arrival time at the end node, which is determined from forward propagation. Equation (A19) is reverse of Equation (A15) to find the feasible transit ranges.

Appendix A.3. Constraint Update

After all the propagation processes are completed, the transit solution \mathbf{t}^k is updated as the new constraints for the next flights:

$$\mathbf{CN}_i^k = [t_e^{k_i} - t_b, t_e^{k_i} + t_b] \quad (\text{A20})$$

$$\mathbf{CL}_j^k = [t_e^{k_j} - t_{lb}, t_e^{k_j} + t_{lb}, t_{x_j}^k - t_{lb}, t_{x_j}^k + t_{lb}] \quad (\text{A21})$$

$$\mathbf{CCN}_i^k = \mathbf{CCN}_i^{k-1} \cup \mathbf{CN}_i^k \quad (\text{A22})$$

$$\mathbf{CCL}_j^k = \mathbf{CCL}_j^{k-1} \cup \mathbf{CL}_j^k \quad (\text{A23})$$

where t_b is a blocking time constraint of a node determined by the rate constraint imposed at nodes, and t_{lb} is the constraint for maintaining along-trail spacing between leading and trailing aircraft on the same link. Figure A2 shows the details of Equations (A20) and (A21).

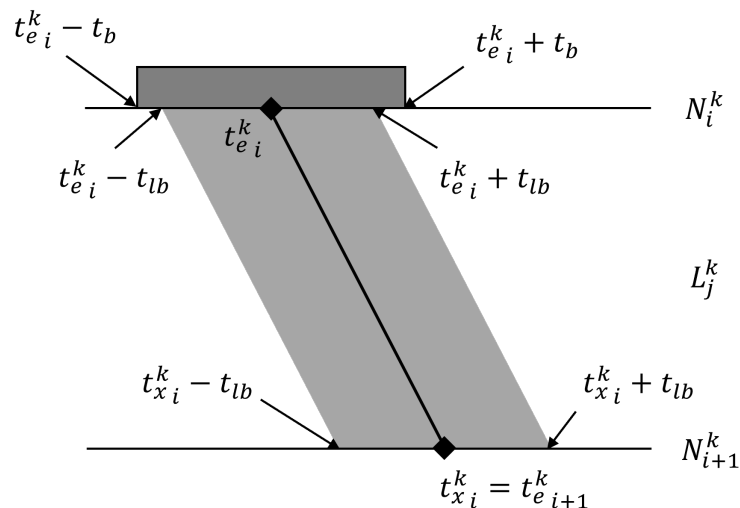


Figure A2. Updated node and link constraints in Equations (A20) and (A21).

References

- Baker, K.R.; Trietsch, D. *Principles of Sequencing and Scheduling*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
- Neuman, F.; Erzberger, H. *Analysis of Sequencing and Scheduling Methods for Arrival Traffic*; NASA Technical Memorandum 102795; NASA Ames Center: Mountain View, CA, USA, 1990.
- Wu, S.; Liu, X. Optimized sequencing and scheduling algorithms for arrival air traffics based on FCFS principles. *IFAC Proc. Vol.* **1994**, *27*, 215–218. [[CrossRef](#)]
- Smeltink, J.W.; Soomer, M.J.; de Waal, P.R.; van der Mei, R.D. An Optimisation Model for Airport Taxi Scheduling. In Proceedings of the INFORMS Annual Meeting, Denver, CO, USA, 24–27 October 2004.
- Roling, P.C.; Visser, H.G. Optimal Airport Surface Traffic Planning Using Mixed-Integer Linear Programming. *Int. J. Aerosp. Eng.* **2008**, *2008*, 732828. [[CrossRef](#)]
- Rathinam, S.; Montoya, J.; Jung, Y. An Optimization Model for Reducing Aircraft Taxi Times at the Dallas Fort Worth International Airport. In Proceedings of the 26th International Congress of the Aeronautical Sciences (ICAS), Anchorage, Alaska, 14–19 September 2008.
- Montoya, J.; Wood, Z.; Rathinam, S.; Malik, W. A Mixed Integer Linear Program for Solving a Multiple Route Taxi Scheduling Problem. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Toronto, ON, Canada, 2–5 August 2010.
- Lee, H.; Balakrishnan, H. A Comparison of Two Optimization Approaches for Airport Taxiway and Runway Scheduling. In Proceedings of the 2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC), Williamsburg, VA, USA, 14–18 October 2012.
- Bosson, C.S.; Sun, D. Optimization of airport surface operations under uncertainty. *J. Air Transp.* **2016**, *24*, 84–92. [[CrossRef](#)]
- Yang, Y.; Gao, Z. Stochastic Scheduling of Ground Movement Problem Integrated with Taxiway Routing and Gate/Stand Allocation. *IET Intell. Transp. Syst.* **2022**, *16*, 1143–1163. [[CrossRef](#)]
- Avella, P.; Boccia, M.; Mannino, C.; Vasilyev, I. Time-Indexed Formulations for the Runway Scheduling Problem. *Transp. Sci.* **2017**, *51*, 1196–1209. [[CrossRef](#)]
- Prakash, R.; Piplani, R.; Desai, J. An Optimal Data-Splitting Algorithm for Aircraft Scheduling on a Single Runway to Maximize Throughput. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 570–581. [[CrossRef](#)]
- Hu, X.; Teng, J.; Wu, W.; Li, Y.; Sheng, Y. Research on Airport Scheduling of FGAP Multi-Objective Programming Model based on Uncertainty Theory. *Symmetry* **2021**, *13*, 1915. [[CrossRef](#)]
- Wang, X.; Brownlee, A.E.; Weiszner, M.; Woodward, J.R.; Mahfouf, M.; Chen, J. A Chance-Constrained Programming Model for Airport Ground Movement Optimisation with Taxi Time Uncertainties. *Transp. Res. Part C Emerg. Technol.* **2021**, *132*, 103382. [[CrossRef](#)]
- Zou, X.; Cheng, P.; Liu, W.; Cheng, N.; Zhang, J. A Two-Stage Taxi Scheduling Strategy at Airports with Multiple Independent Runways. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 165–184. [[CrossRef](#)]
- Ornek, M.A.; Ozturk, C.; Sugut, I. Integer and Constraint Programming Model Formulations for Flight-Gate Assignment Problem. *Oper. Res.* **2022**, *22*, 135–163. [[CrossRef](#)]
- Dönmez, K.; Çetek, C.; Kaya, O. Aircraft Sequencing and Scheduling in Parallel-Point Merge Systems for Multiple Parallel Runways. *Transp. Res. Rec.* **2022**, *2676*, 108–124. [[CrossRef](#)]
- Ma, J.; Delahaye, D.; Sbihi, M.; Scala, P.M. Integrated Optimization of Arrival, Departure, and Surface Operations. In Proceedings of the 8th International Conference for Research in Air Transportation (ICRAT 2018), Castelldefels, Spain, 26–29 June 2018.
- Yan, P.; Liu, M.S.; Jiao, M.H. Receding Horizon Optimization for Dynamic Joint Scheduling of Taxiways and Gates. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 4155–4160.

20. Sun, R.; Li, J.; Niu, K.; Tian, Y.; Xu, C. Research on Joint Ground Movement Optimization based on Bilevel Programming. *Aerospace* **2022**, *9*, 502. [[CrossRef](#)]
21. Xu, B.; Ma, W.; Ke, H.; Yang, W.; Zhang, H. An Efficient Ant Colony Algorithm based on Rank 2 Matrix Approximation Method for Aircraft Arrival/Departure Scheduling Problem. *Processes* **2022**, *10*, 1825. [[CrossRef](#)]
22. Zhang, Y.; He, Q.; Yang, L.; Liu, C. An Improved Tunicate Swarm Algorithm for Solving the MultiObjective Optimisation Problem of Airport Gate Assignments. *Appl. Sci.* **2022**, *12*, 8203. [[CrossRef](#)]
23. Du, W.; Zhu, S.; Tong, L.; Cai, K.; Liang, Z. Robust Gate Assignment to Minimise Aircraft Conflicts. *Transp. B Transp. Dyn.* **2023**, *11*, 2185497. [[CrossRef](#)]
24. Meyn, L.A. A Closed-Form Solution to Multi-Point Scheduling Problem. In Proceedings of the AIAA Modeling and Simulation Technologies (MST) Conference, AIAA, Toronto, ON, Canada, 2–5 August 2010.
25. Meyn, L.A.; Erzberger, H. Airport Arrival Capacity Benefits Due to Improved Scheduling Accuracy. In Proceedings of the AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO), Arlington, VA, USA, 26–28 September 2005.
26. Palopo, K.; Chatterji, G.B.; Lee, H.-T. Interaction of Airspace Partitions and Traffic Flow Management Delay. In Proceedings of the 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA, Fort Worth, TX, USA, 13–15 September 2010.
27. Park, C.; Lee, H.-T.; Meyn, L.A. Computing Flight Departure Times using an Advanced First-Come First-Served Scheduler. In Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA, Indianapolis, ID, USA, 17–19 September 2012.
28. Park, B.; Lee, H.; Lee, H.-T. Extended First-Come First-Served Scheduling Algorithm. *Int. J. Aeronaut. Space Sci.* **2018**, *19*, 509–517. [[CrossRef](#)]
29. Park, B. A Study on Scheduling Algorithms Based on the First-Come First-Served Approach for Airport Surface Operation. Ph.D. Thesis, Inha University, Incheon, Republic of Korea, 2020.
30. Phillips, J.D.; Sadosky, A.V. *Progress in Scheduling Algorithms for a Collaborative Distributed System for Flight Planning*; NASA Technical Memorandum 20210000561; NASA Ames Center: Mountain View, CA, USA, 2021.
31. Park, B.; Lee, H.; Lee, H.-T.; Eun, Y.; Jeon, D.; Zhu, Z.; Lee, H.; Jung, Y. Comparison of First-Come First-Served and Optimization Based Scheduling Algorithms for Integrated Departure and Arrival Management. In Proceedings of the 18th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, AIAA, Atlanta, GA, USA, 25–29 June 2018.
32. Eun, Y.; Jeon, D.; Lee, H.; Jung, Y.C.; Zhu, Z.; Jeong, M.; Kim, H.; Oh, E.; Hong, S. Optimization of Airport Surface Traffic: A Case-Study of Incheon International Airport. In Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference, Denver, CO, USA, 5–9 June 2017.
33. Ministry of Land, Infrastructure, and Transport, Republic of Korea. UBIKAIS. 2015. Available online: <https://ubikais.fois.go.kr:8030/sysUbikais/biz/fpl/dep> (accessed on 11 September 2023).
34. ICAO. *Procedures for Air Navigation and Air Traffic Management (PANS-ATM DOC 4444)*; ICAO: Montreal, QC, Canada, 2016.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.