Comparison of First-Come First-Served and Optimization Based Scheduling Algorithms for Integrated Departure and Arrival Management

Bae-Seon Park[*], Hyeonwoong Lee[†], and Hak-Tae Lee[‡]
Inha University, Incheon, 22212, Republic of Korea

Yeonju Eun[§] and Daekeun Jeon[¶]
Korea Aerospace Research Institute, Daejeon, 34133, Republic of Korea

Zhifan Zhu[‖]
Stinger Ghaffarian Technologies, Moffett Field, CA, 94035, USA

Hanbong Lee[**] and Yoon C. Jung[††]
NASA Ames Research Center, Moffett Field, CA, 94035, USA

Korea Aerospace Research Institute (KARI) and National Aeronautics and Space Administration (NASA) are investigating scheduling algorithms that will be a part of an integrated arrival and departure management system. Inha University, one of the Korean collaborators of KARI, developed an Extended First-Come First-Served (EFCFS) algorithm that is robust and efficient. However, since the EFCFS algorithm sequentially computes the schedule based on priority, the end results may not be optimal for system efficiency. The approach based on Mixed Integer Linear Programming (MILP) originally developed by NASA and modified by KARI is known to produce better schedules at the expense of computational cost. In this paper, the two different scheduling approaches are compared using common traffic scenarios and constraints at Incheon International Airport. Capabilities to apply weight class based wake turbulence runway separation minima and Miles-in-Trail (MIT) restrictions at selected meter fixes are added to the previously developed EFCFS scheduler. Based on historic data, 40 departures and 20 arrivals are chosen in a one-hour period and 100 scenarios were created by randomly assigning gate numbers, gate departure times, and runway landing times. With the current runway separation requirements, MILP resulted in about ten to twenty percent smaller average delays depending on the constraints. With artificially increased separation minima, the difference between MILP and EFCFS became more noticeable. However, the EFCFS was about ten times faster with smaller variations among different scenarios and constraints. The comparison suggests that the MILP-based algorithm has a small advantage at the current traffic level; however, has potential to be more effective in higher demand or severe weather situations. The EFCFS algorithm may be better suited for real-time applications or investigating larger scale scheduling problems.

I. Introduction

Iɴ Korea, an integrated departure and arrival management system is under development to reduce congestion and increase efficiency at busy airports. The research and development effort is funded by the Ministry of Land,

---

[*]Ph.D. Student, Dept. of Aerospace Engineering, 100 Inha-ro, Nam-gu, Incheon 22212, Republic of Korea, AIAA Student Member

[†]M.S. Student, Dept. of Aerospace Engineering, 100 Inha-ro, Nam-gu, Incheon 22212, Republic of Korea, AIAA Student Member

[‡]Associate Professor, Dept. of Aerospace Engineering, 100 Inha-ro, Nam-gu, Incheon 22212, Republic of Korea, AIAA Member

[§]Senior Researcher, CNS/ATM Team, 169-84 Gwahak-ro, Yuseong-gu, Daejeon 34133, Republic of Korea

[¶]Principal Researcher, CNS/ATM Team, 169-84 Gwahak-ro, Yuseong-gu, Daejeon 34133, Republic of Korea

[‖]Senior Software Engineer, Aerospace High-Density Operations Research Branch, Mail Stop 210-8, Moffett Field, CA 94035, USA

[**]Aerospace Engineer, Aerospace High-Density Operations Research Branch, Mail Stop 210-6, Moffett Field, CA 94035, USA, AIAA Member

[††]Aerospace Engineer, Aerospace High-Density Operations Research Branch, Mail Stop 210-6, Moffett Field, CA 94035, USA, AIAA Senior Member

Infrastructure, and Transportation (MOLIT) and performed by a consortium led by KARI consisting of universities, research institutions, and private companies. Aircraft scheduling algorithms are one of the key components of this system. Inha University, a member of the consortium, is developing a scheduler based on a First-Come First-Served (FCFS) approach [1, 2]. KARI is developing a scheduler using MILP through collaborative research with NASA [3, 4].

The Extended FCFS (EFCFS) scheduler can solve any scheduling problem that can be represented by a node-link structure. This algorithm was originally developed to solve arrival metering problems [5]. Later, link constraints were added to solve Traffic Flow Management problems [6]. To apply this algorithm to aircraft surface movement problems at airports, constraints for link directionality and junction nodes were added, and the algorithm was tested using historic flight data and a node-link model of Incheon International Airport (ICN) [1, 2]. Figure 1 shows the node-link model of ICN used for the current study.
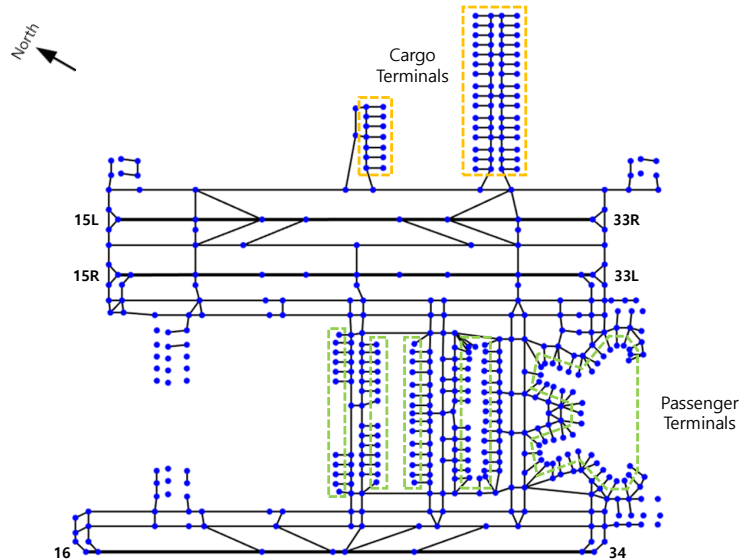

Fig. 1   Node-link model of Incheon International Airport.

MILP-based optimization models were developed separately for runway scheduling and taxiway scheduling. The two MILP-based schedulers are linked through common requirements such as earliest possible off-block times and target take-off sequence and times. Runway scheduling, which was originally developed by NASA [7], was modified to incorporate a new restriction for smooth merging of departure flights into overflight traffic streams. Taxiway scheduling computes the optimal pushback times that follow the desired take-off sequence and times from the runway scheduler and minimize the taxi times. In this taxiway scheduling, a node-link model with predetermined taxi routes is used, and constraints such as conflict-free conditions among all taxi-in and taxi-out aircraft are considered [4].

In this paper, two schedulers using different approaches are compared, which is one of the first attempts to systematically compare the two scheduling approaches. Common scenarios and constraint sets are established for the comparison. Resulting delays and computation times are compared.

Following this introduction, Section II briefly explains the two different scheduling approaches. Section III presents the common traffic scenarios for ICN and compares the scheduling results. Section IV concludes the paper.

II. Scheduling Approaches

A. Extended First-Come First-Served Approach

Flight progression can be represented as a sequence of flight paths and simplified to a node-link structure. If a scheduling problem can be formulated into a node-link structure, the EFCFS scheduler can determine the earliest arrival and departure time of each flight with the given scheduling priority [1, 2, 6].

The EFCFS scheduling process consists of two propagation steps [1, 6]. For a given flight in the scheduling priority, an initial forward propagation step determines the earliest finish times satisfying all constraints at nodes and links, then

2

backward propagation computes the earliest start time of the given flight. For departure flights, gate departure time is the start time and takeoff time is the finish time. For arrival flights, landing time is the start time and gate arrival time is the finish time. As EFCFS can vary taxi speed within a given range, it determines the required taxi speed at each link during the backward propagation step. When the schedule of the given flight is fixed, all nodes and links that the flight transits are updated. The scheduler repeats the same process for the next flight in a priority order.

The EFCFS scheduler can evaluate multiple taxi routes and select the best route for a given flight. In one of the previous studies [2], several cases were found where one of the alternate routes was chosen by the scheduler. However, this capability is not used for the current study.

The EFCFS is also different from conventional FCFS approaches in several ways. In the EFCFS algorithm, the term 'come' means the scheduling priority. If the scheduling priority is set to the original scheduled departure or arrival times, it is equivalent to the conventional 'first-come'. The term 'served' in EFCFS means freezing of the schedule rather than the actual departure or arrival. EFCFS is capable of producing a solution that switches the departure or arrival orders. So 'first-served' can be considered 'first-scheduled'. The EFCFS still retains the fundamental characteristics of the conventional FCFS, which sequentially calculates the schedule of each flight.

B. Optimization Based Approach

Airport surface traffic optimization for ICN has been studied with the requirements and the relevant assumptions defined based on the airport operational characteristics [3]. For the surface traffic optimization, the three-step approach [7–10] was applied, which consists of; Step 1) unimpeded taxi-out time estimation; Step 2) runway scheduling for departures; and Step 3) taxiway scheduling. The MILP was used for runway scheduling and taxi scheduling in Step 2 and 3 in this study.

In runway scheduling, Traffic Management Initiatives (TMI) restrictions such as Miles-in-Trail (MIT), Minimum Departure Interval (MDI), Expected Departure Clearance Time (EDCT), and Call For Release (CFR) were all incorporated. In addition, for some special restrictions on the west-bound departures from multiple runways in ICN, a new type of TMI was also considered. It comes from the multiple available time slots, where a departure aircraft is allowed to merge into the overhead stream avoiding the expected times of the other overflight traffic at the merging fix. They were converted into the available time slots for take-off using the departure transit times from runway to the merging fix, and referred to as 'multiple take-off time windows' [4].

For the TMI-type requirements on the shared departure fixes of the multiple runways in ICN, the runway scheduling problem has been formulated into a single MILP model using appropriate separations between any pair of arrival, departure, and crossing flights on both runways. An alternative approach was also considered for computational efficiency. In the alternative approach, two runway scheduling problems were solved sequentially. That is, the runway scheduling problem of a certain runway was solved first and then, in the second runway scheduling problem, the calculated take-off times of the departures from the first runway to the shared departure fix were applied as constraints for the departures from the other runway using the 'multiple take-off time windows' constraint. The Monte Carlo test results of the previous work [4] showed that the alternative approach to implementing the two sequential optimizations had the better computation time performance with compromised cost, compared to the single MILP model for the multiple runway scheduling problem.

Taxiway scheduling was formulated for the purpose of determining the optimal pushback times for departures, which follow the desired take-off times from the runway scheduling results, considering operational and safety related constraints. The MILP model of the taxi scheduling problem incorporates passage times at all intersections along the taxi routes as decision variables, and also includes appropriate constraints to maintain safe separation between aircraft moving on the surface. In the taxiway scheduling, the cost function is formulated aiming to minimize the late take-off times compared to the desired take-off times from the runway scheduling results, and the sum of taxi-in and taxi-out times. For this, all the runway operations were considered in the taxiway scheduling problem. Active runway crossing constraints that may cause taxi-out time extension were also incorporated [4].

C. Compatibility of the Two Algorithms

For the current study, the EFCFS uses the same predetermined routes that the MILP scheduler uses. In the input scenarios, the landing times of arrivals are pre-adjusted to satisfy the wake turbulence separation minima. The MILP scheduler performs only the taxi scheduling for the arrival flights. To model this, the EFCFS scheduler used arrival priority. The schedules of all the arrival flights are fixed first, and then departure flights are scheduled.

Table 1 lists all the common constraints implemented to both schedulers to solve the surface and terminal area

scheduling problems. The first three constraints had already been implemented in the EFCFS scheduler in previous studies [1, 2].

Table 1   List of common constraints.

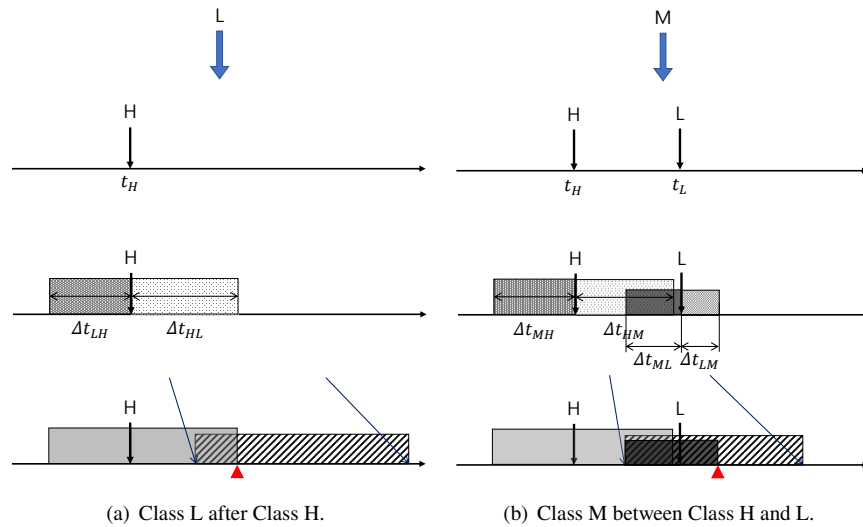| |
| --- |
| Earliest possible pushback times of departures |
| No deadlock in bi-directional taxiway links |
| Aircraft separation along the taxiways |
| Runway separation based on aircraft weight class |
| Miles-In-Trails (MIT) |

(a) Class L after Class H.   (b) Class M between Class H and L.

Fig. 2   Different sizes of time blocks determined by aircraft wake turbulence categories (H, M, and L).

Two enhancements were made to the EFCFS scheduler to handle the last two constraints from Table 1. The first enhancement is to apply the runway separation minima based on aircraft wake turbulence category (WTC) because the minimum runway separations are dependent upon the weight classes of consecutive aircraft. Instead of blocking time intervals calculated based on given departure or arrival rates [2], the scheduler dynamically recalculates the blocked time slots based on the WTC of the aircraft that have already been scheduled and the WTC of the aircraft that is being scheduled. In Fig. 2(a), a heavy (H) class aircraft has already been scheduled at time t_H, and a light (L) class aircraft is being scheduled. If Δt_{ij} denotes the separation minimum between leading i class aircraft and trailing j class aircraft, the size of the blocked time slot should be Δt_{LH} in front of t_H. Similarly, the size of the blocked time behind t_H should be Δt_{HL}. The total blocked time interval is $[t_H - \Delta t_{LH}, t_H + \Delta t_{HL}]$ where the L class aircraft cannot be scheduled due to the existing H class aircraft. If the possible landing time interval is shown as the hatched area, the earliest time that is outside the blocked time slot denoted by the red triangle becomes the scheduled arrival time of this L class aircraft. Figure 2(b) illustrates a more complicated situation where an H class and an L class aircraft have already been scheduled at t_H and t_L, respectively, and a medium (M) class aircraft is being scheduled. Around t_H, the blocked time interval should be $[t_H - \Delta t_{MH}, t_H + \Delta t_{HM}]$. Around t_L, the blocked time interval should be $[t_L - \Delta t_{ML}, t_L + \Delta t_{LM}]$. The union set of the two intervals represents the final blocked time slot for the M class aircraft in this situation.

The second enhancement is handling the MIT constraints. Since the EFCFS was originally developed for metering problems in the terminal area [5] and is based on the available and unavailable time slots, MIT constraints can naturally be added by extending the node-link from the runway to the metering fix and adding propagation steps with proper time slots as shown in Fig. 3. Since MIT constraints are applied, the distance requirements are converted to blocked time intervals by dividing the MIT distance by the expected speed of the aircraft.
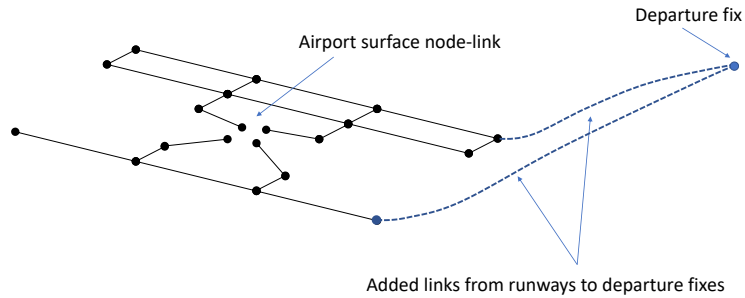
Fig. 3 Extra node-link for departure fix.

III. Scheduling Results

A. Problem Set

Based on the departure peak time data in April 2015, 40 departures and 20 arrivals within a one-hour time window were selected for test traffic scenarios. With the number of flights fixed, gate numbers, gate departure times, and runway landing times were randomly chosen to generate multiple scenarios. Among those scenarios, 100 selected scenarios had runway landing times that already satisfied the WTC separation criteria. The schedulers do not reschedule the runway landing times.
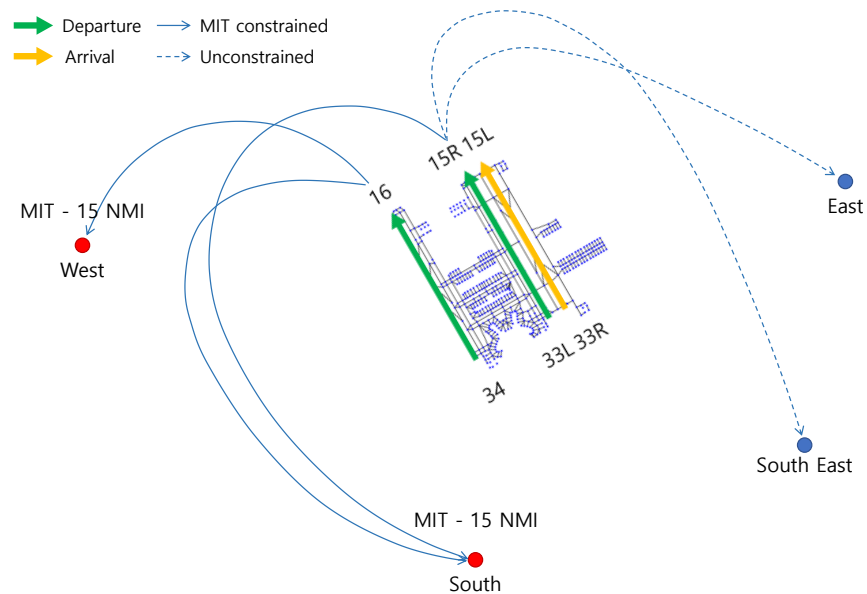

Fig. 4 Runway directions and departure fixes at ICN.

Departure fixes consist of the West, South, South East, and East departure fixes as shown in Fig. 4. South East and East fixes are unconstrained. Five aircraft taking off from runway 15R/33L transit each fix. The MIT constraint of 15 nmi is applied to the West and South fixes. Among the departures from runway 16/36, 15 departures transit the West fix after takeoff. In addition, 9 aircraft taking off from runway 16/34 and 6 aircraft taking off from runway 15R/33L merge at the South fix.

In these scenarios, only two WTCs, medium and heavy, were considered. The mixture of the fleet in all scenarios is shown in Table 2. All 20 arrivals are assigned to runway 15L/33R. For departures, 16 aircraft are assigned to runway 15R/33L, and 24 to runway 16/34. Runways 15L/33R and 15R/33L are considered dependent. The runway separation minima are shown in Table 3. The separation minima for the departure after the adjacent runway arrival was considered zero.

5

Table 2   Fleet mix for all scenarios.

WTC / State	Departure	Arrival
Medium	14	7
Heavy	26	13

Table 3   Runway separation minima in seconds.

(a) Departure after same runway departure.

Lead / Trail	Medium	Heavy
Medium	120	120
Heavy	180	120

(b) Arrival after same runway departure.

Lead / Trail	Medium	Heavy
Medium	52	45
Heavy	52	45

(c) Departure after same runway arrival.

Lead / Trail	Medium	Heavy
Medium	60	50
Heavy	60	50

(d) Arrival after same runway arrival.

Lead / Trail	Medium	Heavy
Medium	120	120
Heavy	120	120

(e) Arrival after adjacent runway departure.

Lead / Trail	Medium	Heavy
Medium	52	45
Heavy	52	45

For taxi routes, the most commonly used route for each aircraft based on historical data was selected as the default route. The aircraft separation distance on the taxiway links is assumed to be 80m. The maximum taxi speeds are set to 5 knots for pushback links, 10 knots for ramp links, and 15 knots for taxi links. For the EFCFS scheduler, once an aircraft starts moving, it will continue to the final destination node without stopping. Taxi speed can vary from 90 percent of the maximum speed up to the maximum speed designated for the link. For the MILP scheduler, any speed from zero to the maximum speed can be used.

Departure aircraft are assumed to use the entire length of the runway from the threshold node to the opposite threshold node. Arrival aircraft land at threshold nodes and use the runway from the threshold node to the runway exit node. Other taxiing aircraft cannot cross the runway while aircraft are taking off or landing. The runway occupancy times are shown in Table 4. Both the schedulers do not adjust the transit times of the aircraft once they leave the airport surface. Traffic situations before and after the scheduling period as well as overflights around the airport are not considered in this study.

Table 4   Runway occupancy times in seconds.

WTC / State	Departure	Arrival	Cross
Medium	57	52	30
Heavy	50	45	30

B. Scheduling Results

1. Without MIT Constraints

For scheduling without MIT constraints, departure aircraft are scheduled until they reach the beginning of the runways. Runway landing times that already satisfy the WTC separation minima are not adjusted, and arrival priority is used for the EFCFS algorithm. Since the EFCFS computes the runway and taxi schedules at once, this resulted in zero runway and taxi delays for arrival flights.

Figure 5 shows the gate departure and takeoff delay distributions for a total of 4,000 departing aircraft accumulated from scheduling over 100 scenarios. For the gate departure, the average delays of EFCFS and MILP are 2.8 minutes and

2.6 minutes, respectively, where there are no significant differences as shown in Fig. 5(a).

Figure 5(b) shows the takeoff delay distribution for departure flights. EFCFS has more flights in the 3-10 minute delay range. The takeoff delay is based on the unimpeded taxi time from the departure gate to runway threshold at the maximum taxi speed. The average takeoff delay of the EFCFS is 3.2 minutes, whereas the average delay using the MILP is 2.6 minutes. It shows a larger difference between the two schedulers than the gate departure delay because the EFCFS seeks the earliest gate departure time within the given taxi speed range so that the gate delay is relatively smaller than the taxi delay.

(a) Gate departure delay.

(b) Takeoff delay.

Fig. 5   Delay distributions for departures of total 100 scenarios.
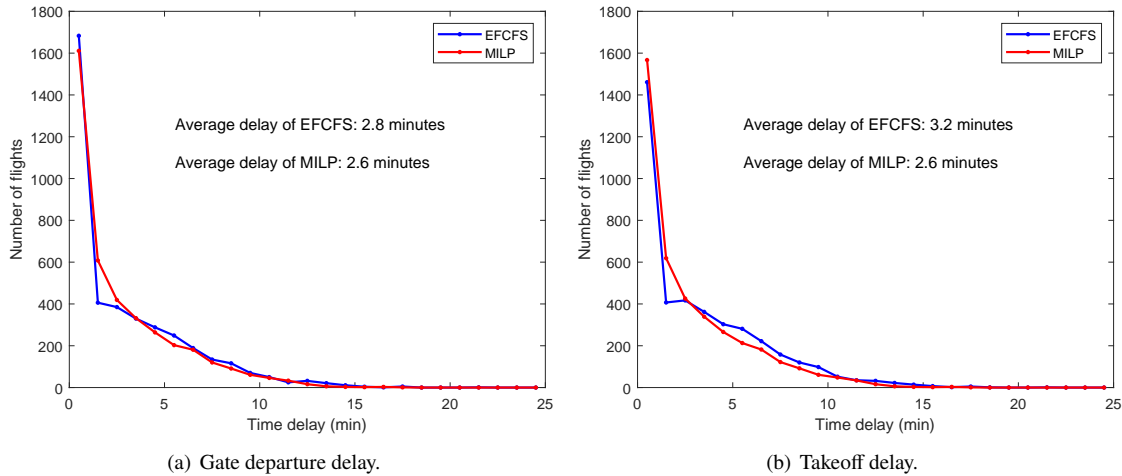
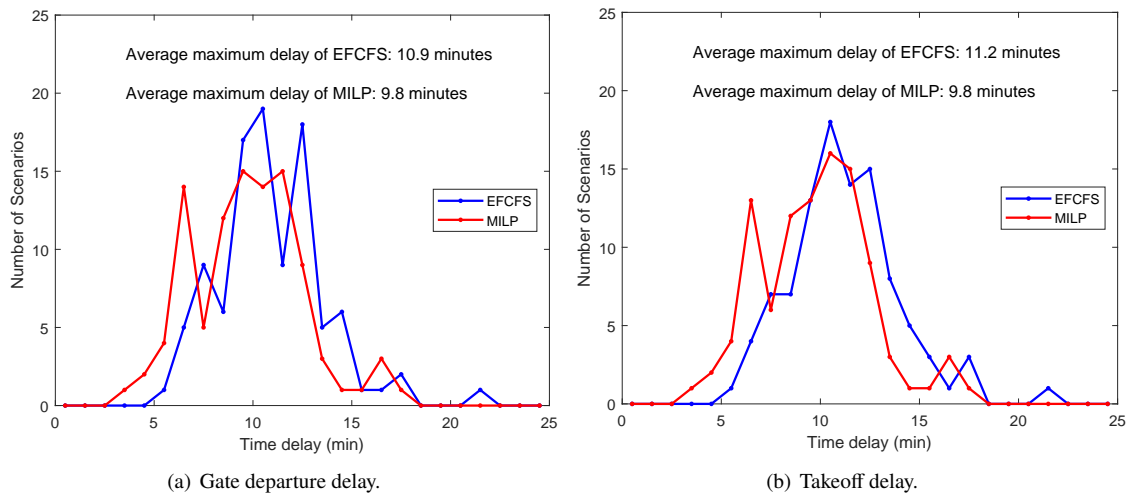(a) Gate departure delay.

(b) Takeoff delay.

Fig. 6   Maximum delay distributions for departures of total 100 scenarios.

Figure 6 shows the maximum gate departure delay distributions and the maximum takeoff delay distributions for the 100 scenarios. Average maximum gate departure delays of EFCFS and MILP are 10.9 minutes and 9.8 minutes, respectively. Average maximum takeoff delays of EFCFS and MILP are 11.2 minutes and 9.8 minutes, respectively. The MILP scheduler shows slightly better performance as can be seen from the distribution in Fig. 6 (b). The maximum takeoff delay distribution curve of the EFCFS scheduler is shifted to the right side compared to the distribution curve of the MILP algorithm. The average value of the maximum gate arrival delays for the MILP is 1.8 minutes, with the maximum value of 6 minutes.
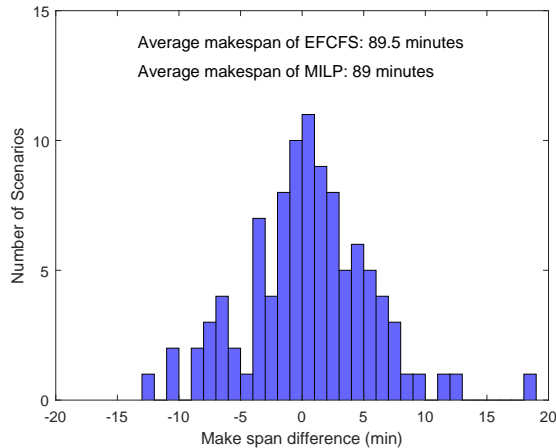
Fig. 7 Makespan differences between EFCFS and MILP for 100 scenarios.

Figure 7 shows the distribution of makespan differences between the two schedulers. In Fig. 7, the positive difference means that EFCFS has a larger makespan than MILP. The average makespans of EFCFS and MILP are 89.5 minutes and 89 minutes, respectively. The MILP model shows slightly better performance, but the difference is small.

2. With MIT Constraints

For scheduling including MIT constraints, departure aircraft are scheduled until they reach the departure fixes. The 15 nmi MIT constraint was applied to the West and South departure fixes. Figure 8 shows the gate departure and takeoff delay distributions. Average gate departure delays of EFCFS and MILP are 3.3 minutes and 3.2 minutes, respectively. The EFCFS scheduler generated more flights that have delays within one minute, similar to the no MIT results. The average takeoff delays are 3.7 minutes and 3.3 minutes for EFCFS and MILP, respectively. The EFCFS approach has more flights with delays in the range of 5-10 minutes, whereas the MILP approach has more flights with less than a one-minute delay. With the MIT constraints, the average delay differences between EFCFS and MILP become smaller. It is speculated that even though the MILP scheduler can maximize the runway throughput, it becomes less effective due to the additional constraints at the fixes.
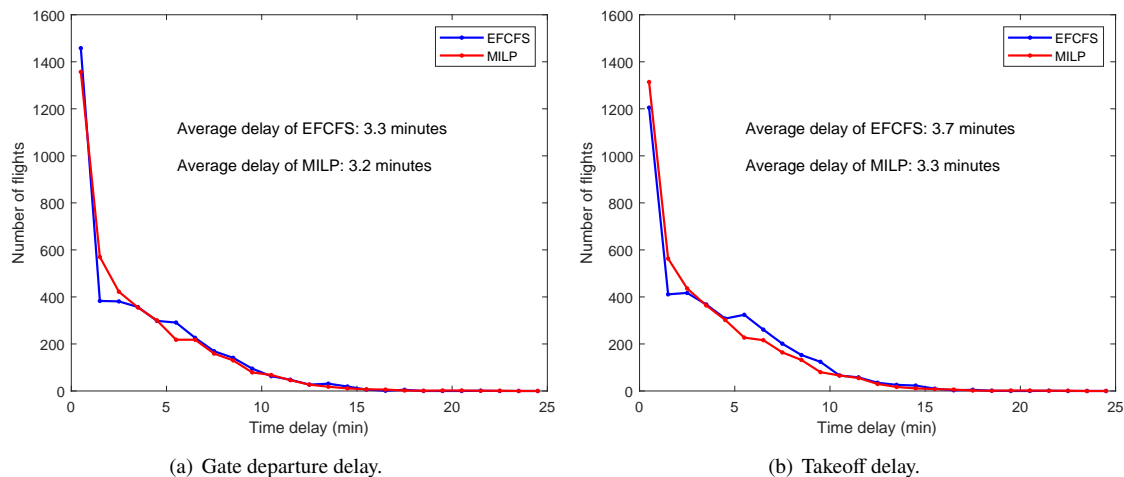

(a) Gate departure delay.

(b) Takeoff delay.

Fig. 8 Delay distributions for departures of total 100 scenarios.

Figure 9 shows the maximum gate departure delay and takeoff delay distributions for all scenarios. The maximum delays of both schedulers are less than 25 minutes. There is no notable difference between the two scheduling approaches

8

in terms of the maximum delay.
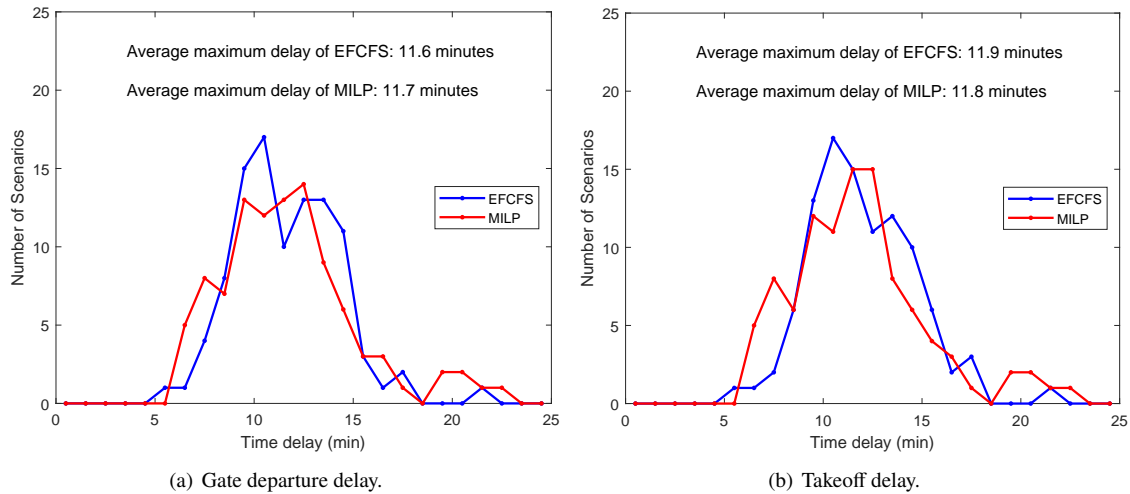

(a) Gate departure delay.

(b) Takeoff delay.

Fig. 9 Maximum delay distributions for departures of total 100 scenarios.

Figure 10 shows the distribution of makespan differences between the two schedulers. The average makespans are 90.2 minutes for EFCFS and 89.7 minutes for MILP. Again, the MILP scheduler showed slightly better performance, but the difference is very small.
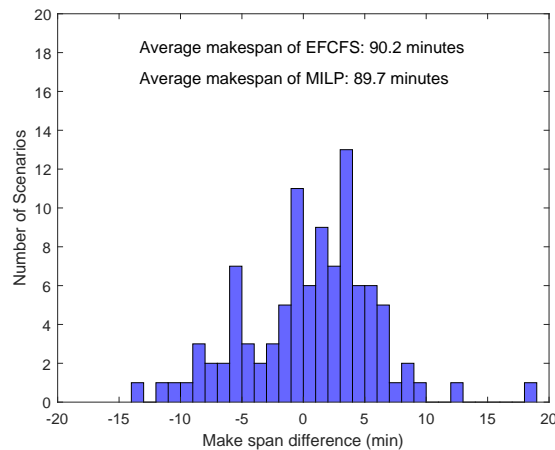

Fig. 10 Makespan differences between EFCFS and MILP for 100 scenarios.

Since the landing times of arriving flights were considered not adjustable, the arrival runway delays using both schedulers were zero. Since the EFCFS use arrival priority that schedules all the arrival flights from runways to gates, no taxi delays were added by the scheduler. The MILP model generated an average 0.2 minutes of taxi delay from runways to the arrival gates regardless of the MIT constraints.

3. Adjusted Runway Separation Minima

The MILP scheduler shows slightly better performance than the EFCFS in the previous two results. To identify the differences and causes more clearly, additional scheduling was performed with artificially increased runway separation minima to significantly increase delay. To increase the impact of takeoff sequencing, the minimum separation was increased to five minutes from the original two-minute separation and ten minutes from the original three minutes in Table 3. In this case, MIT is not applied because the aircraft are expected to maintain sufficient intervals at the departure fixes. Note that this does not model the severe weather operation conditions at ICN.
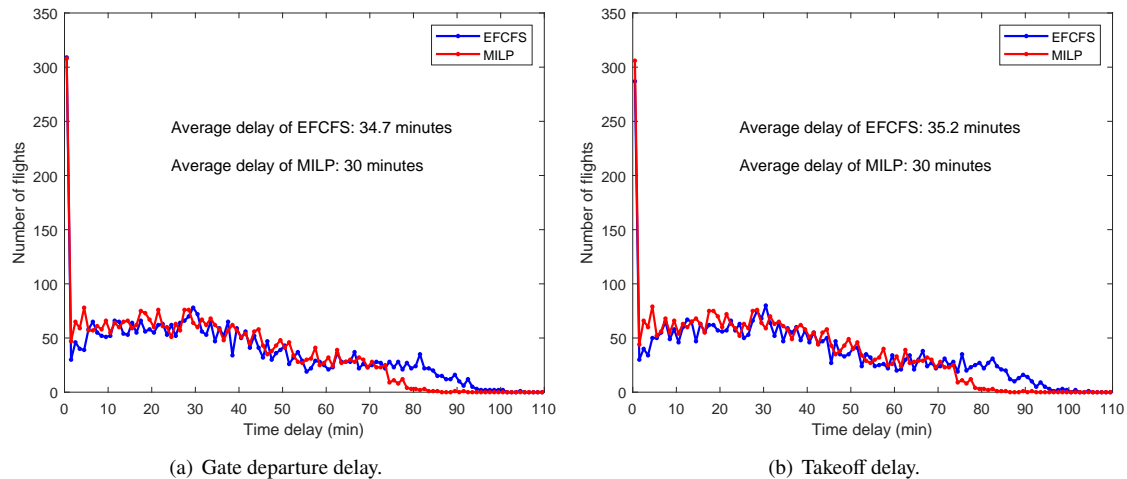
9

(a) Gate departure delay. (b) Takeoff delay.

Fig. 11   Delay distributions for departures of total 100 scenarios.

The scheduling results show that the average gate departure and takeoff delays of the MILP model are both 30 minutes. The EFCFS scheduler produced larger average delays of 34.7 and 35.2 minutes for gate departure and takeoff respectively. Particularly, using EFCFS has more flights with delay larger than 70 minutes compared with using the MILP, as shown in Fig. 11.

For the maximum gate departure and takeoff delays, EFCFS curves are shifted to the right as shown in Fig. 12. Therefore, the EFCFS clearly has larger maximum delays. The average maximum gate departure delays are 73.2 and 87.3 minutes for MILP and EFCFS, respectively. The average maximum takeoff delays are 73.2 and 87.8 minutes for MILP and EFCFS, respectively.
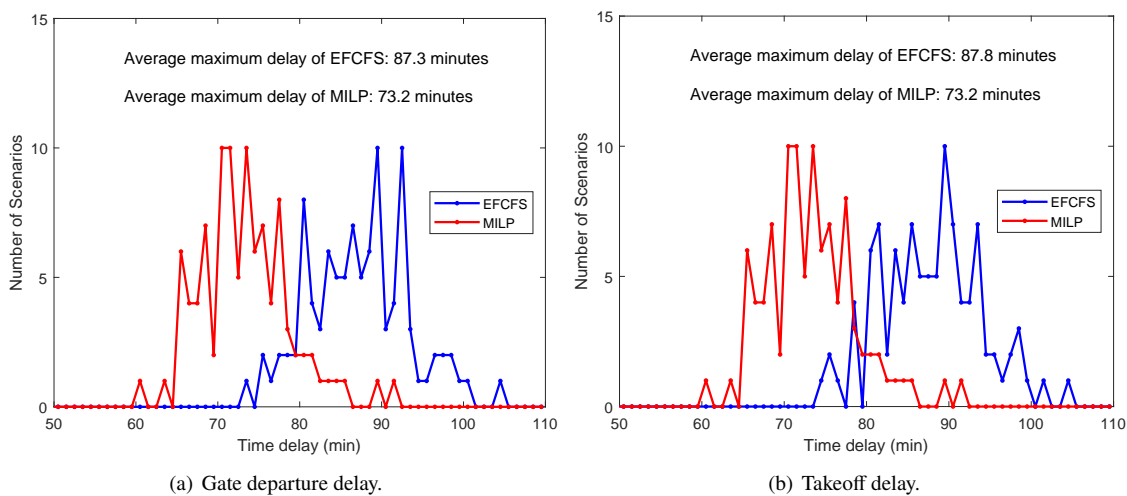

(a) Gate departure delay. (b) Takeoff delay.

Fig. 12   Maximum delay distributions for departures of total 100 scenarios.

The average makespans of both the schedulers are 154.1 and 169.5 minutes for MILP and EFCFS, respectively. Figure 13 shows that most makespan differences are biased in the positive direction, indicating that the MILP is also better in terms of makespan.

When the separation minima are artificially increased, the advantage of MILP becomes clearer. The results suggest that the difference is small for the current level of traffic at ICN. However, the difference is very likely to grow if the airport became more congested in the future.
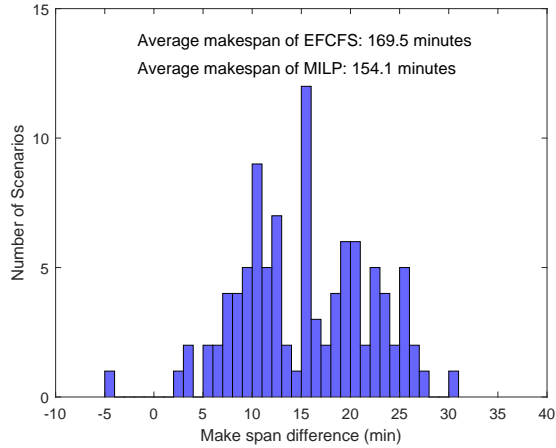
Fig. 13   Makespan differences between EFCFS and MILP for 100 scenarios.

C. Computation Times

To compare the computational performance of EFCFS and MILP, the computation times of the two schedulers for each scenario were measured on a desktop computer with an Intel i7-6820HQ processor running at 2.79GHz and 32GB of RAM, which are plotted in Fig. 14. Figure 14(a) shows the computation times of scheduling without MIT constraints for each scenario. The average computation time of EFCFS is 0.82 seconds whereas that of MILP is 6.39 seconds, about 8 times larger than EFCFS. Also, whereas the EFCFS shows small variations between scenarios, the MILP shows large variations in computation times. Figure 14(b) shows the computation times of scheduling with MIT constraints for each scenario. The average computation times are 0.99 and 9.22 seconds for EFCFS and MILP, respectively. For the MILP model, adding MIT constraints significantly increased the computation time, whereas the EFCFS scheduler showed no notable differences. For the given size of the problem, the EFCFS is about ten times faster. However, this gap can become larger if the size of the problem such as number of aircraft and constraints increase.
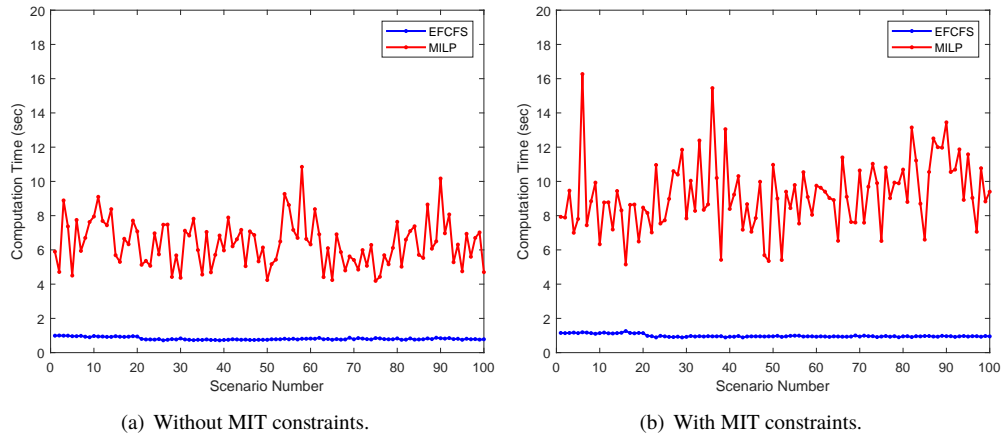

(a) Without MIT constraints.

(b) With MIT constraints.

Fig. 14   Computation times of two schedulers.

Table 5 summarizes the comparison results between the MILP and EFCFS schedulers described in this section.

IV. Conclusions

In this paper, the performances of two different schedulers, MILP and EFCFS, were compared using common scenarios and constraints. With the scenarios based on historic data combined with the separation minima that are currently being used, the MILP approach showed slightly less delays at a cost of ten times the computational time.

11

Table 5   Summary of the scheduling results.

	Without MIT					With MIT					Large separation (without MIT)		
	Takeoff delay (minutes)		Average makespan (minutes)	Computation time (seconds)		Takeoff delay (minutes)		Average makespan (minutes)	Computation time (seconds)		Takeoff delay (minutes)		Average makespan (minutes)
	Average	Average maximum		Average	Standard deviation	Average	Average maximum		Average	Standard deviation	Average	Average maximum	
MILP	2.6	9.8	89	6.39	1.35	3.3	11.8	89.7	9.22	2.03	30	73.2	154.1
EFCFS	3.2	11.2	89.5	0.82	0.08	3.7	11.9	90.2	0.99	0.09	35.2	87.8	169.5

With and without the MIT constraints, the similar results could also be considered as cross verification of the two schedulers. The differences in performance became clearer when the separation minima were artificially increased to cause significant delays. From these results, therefore, it seems that the MILP scheduler, which adjusts the schedule of all the aircraft to minimize system delay, is better suited when trying to reduce delays as much as possible in increased demand or decreased capacity situations. On the other hand, the EFCFS scheduler looks better suited for situations where consistent fast computation speed is required such as real-time operations or large number of aircraft are scheduled. In addition to testing more scenarios, handling of uncertainty is one of the major future research goals.

Acknowledgments

References

[1] Park, B., and Lee, H., “Development of a First-Come First-Served Departure Scheduler,” *the 7th Asia-Pacific International Symposium on Aerospace technology (APISAT)*, Cairns, AU, 2015.

[2] Park, B., Lee, H., Kang, S., and Lee, H., “Airport Surface Movement Scheduling with Route Assignment Using First-Come First-Served Approach,” *the 17th AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, Denver, CO, 2017.

[3] Eun, Y., Jeon, D., Lee, H., Zhu, Z., Jung, Y., Jeong, M., Kim, H., Oh, E., Hong, S., and Lee, J., “Operational Characteristics Identification and Simulation Model Validation for Incheon International Airport,” *the 16th AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, Washington, DC, 2016.

[4] Eun, Y., Jeon, D., Lee, H., Jung, Y., Zhu, Z., Jeong, M., Kim, H., Oh, E., and Hong, S., “Optimization of Airport Surface Traffic: A Case-study of Incheon International Airport,” *the 17th AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, Denver, CO, 2017.

[5] Meyn, L., “A Closed-Form Solution to Multi-Point Scheduling Problems,” *AIAA Modeling and Simulation Technologies Conference*, Toronto, ON, 2010.

[6] Park, C., Lee, H., and Meyn, L., “Computing Flight Departure Times Using an Advanced First-Come First-Served Scheduler,” *AIAA Aviation Technology, Integration, and Operation (ATIO) Conference*, Indianapolis, IN, 2012.

[7] Malik, A. W., Lee, H., and Jung, Y., “Runway Scheduling for Charlotte Douglas International Airport,” *the 16th AIAA Aviation Technology, Integration and Operations (ATIO) Conference*, Washington, DC, 2016.

[8] Lee, H., “Airport Surface Traffic Optimization and Simulation in the Presence of Uncertainties,” Ph.D. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2014.

[9] Lee, H., and Balakrishnan, H., “A Comparison of Two Optimization Approaches for Airport Taxiway and Runway Scheduling,” *the 31st Digital Avionics Systems Conference (DASC)*, Williamsburg, VA, 2012.

[10] Balakrishnan, H., and Jung, Y., “A Framework for Coordinated Surface Operations Planning at Dallas-Fort Worth International Airport,” *AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, SC, 2007.