공학박사 학위논문

Robust Aircraft Scheduling in Point Merge System Under Uncertainties

불확실성을 고려한 포인트 머지 시스템에서의
강건 항공기 스케줄링

2021 년 2 월

서울대학교 대학원
기계항공공학부

이 소 망

Robust Aircraft Scheduling in Point Merge System Under Uncertainties

불확실성을 고려한 포인트 머지 시스템에서의
강건 항공기 스케줄링

지도교수 김 유 단

이 논문을 공학박사 학위논문으로 제출함

2020 년 12 월

서울대학교 대학원

기계항공공학부

이 소 망

이소망의 공학박사 학위논문을 인준함

2020 년 12 월

위 원 장 기 창 돈 (인)

부위원장 김 유 단 (인)

위 원 김 현 진 (인)

위 원 박 찬 국 (인)

위 원 이 금 진 (인)

Robust Aircraft Scheduling in Point Merge System Under Uncertainties

by

Somang Lee

Submitted to the Graduate School of
Seoul National University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical and Aerospace Engineering

Seoul National University

Advisor: Prof. Youdan Kim

February 2021

Robust Aircraft Scheduling
in Point Merge System Under Uncertainties

by

Somang Lee

Approved as to style and content by:

Changdon Kee, Chair, Ph.D.

Youdan Kim, Vice-Chair, Ph.D.

H. Jin Kim, Member, Ph.D.

Chan Gook Park, Member, Ph.D.

Keumjin Lee, Member, Ph.D.

Abstract

Robust Aircraft Scheduling in Point Merge System Under Uncertainties

Somang Lee
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

This dissertation reports on the scheduling algorithms in the Point Merge System (PMS). Specifically, two problems are addressed: a scheduling problem without uncertainties, including a holding pattern, and a robust scheduling problem considering uncertainties.

In the PMS, the scheduling algorithm including a holding pattern is described based on the previous result, a PMS scheduling algorithm. The PMS configuration is transformed to a node–link structure, and the optimization problem is formulated using binary and integer variables. The conventional Mixed Integer Linear Programming (MILP) formulation cannot be directly applied to the transformed node–link structure because of the characteristics of holding patterns. Thus, to apply the MILP formulation to the proposed PMS holding algorithm, the route structure is changed, and virtual fixes are introduced. Several suitable constraints such as first-in-first-out and discrete holding delay constraints are introduced to incorporate the characteristics of the holding procedure in the MILP formulation.

Moreover, the robust scheduling algorithm of the PMS is provided considering the uncertainties of the Estimated Time of Arrival (ETA) and Continuous Descent Approach (CDA). The methodology of robust optimization is described. Because the uncertain constraint can be converted to a deterministic constraint through the chance constraint, the ETA and CDA uncertainty models are assumed to follow a normal distribution. In particular, the dependence of the ETA uncertainty on the remaining flight time, and the correlation between the previous and subsequent uncertainties are reflected through a multivariate normal distribution. The sliding window technique is used to reduce the computation load in the numerical simulation.

The numerical simulations demonstrate the performance of the proposed scheduling algorithms. The Monte Carlo simulation results show that the PMS algorithm considering a holding pattern exhibits a high aircraft capability and an additional degree of freedom. Moreover, the performance of the robust scheduling algorithm is evaluated in terms of the number of constraint violations, average delay, and amount of schedule change. The Monte Carlo simulation indicates that the robust scheduling algorithm corresponds to fewer constraint violations and reduces the amount of schedule change.

Keywords: Aircraft Sequencing and Scheduling, Robust Optimization, Estimated Time of Arrival (ETA), Point Merge System (PMS), Uncertainty, Holding Pattern

Student Number: 2014-21893

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

In recent years, air traffic has increased and Air Traffic Management (ATM) has received considerable attention. According to the Air Transport Action Group, the number of total air passengers worldwide is expected to double from 3.3 billion in 2014 to 6.9 billion in 2034. [1] With the airspace near the airport becoming saturated, the amount of airborne delays is increasing, along with the workload of human air traffic controllers to manage the growing air traffic.

To alleviate these problems, several procedures have been recommended and adopted in most airports, for example, Standard Instrument Departure (SID), which is a predefined departure process for a flight, and Standard Terminal Arrival Route (STAR), which is an arrival procedure in the Terminal Maneuvering Area (TMA). These procedures were developed to accommodate as many aircraft categories as possible, thereby reducing the complexity of communication between the pilot and air traffic controller. [2] However, if the airspace is congested to a level beyond the developed procedure, the air traffic controller must manage the flights using other approaches, such as radar vectoring. Neverthe-

less, such traditional approaches involve a higher workload for the air traffic controllers as they must provide the appropriate heading, altitude, and speed direction to each flight. In this case, the communication load of the air traffic controllers also increases.

Many researchers have attempted to develop a decision support tool for air traffic controllers. Single European Sky ATM Research system (SESAR) of Europe and the Next Generation Air Transportation System (NextGen) of the USA represent the latest efforts to alleviate the workload of air traffic controllers by using schedule managing tools. [3]

In addition, the point merge system (PMS) has been recently developed to efficiently manage the arrival flow of aircraft. [4] The PMS, proposed by EUROCONTROL Experimental Centre in 2006, has been widely adopted by many airports including the Oslo, Dublin, and Jeju international airports. Compared to radar vectoring, the PMS can simplify the tasks of air traffic controllers and reduce the communication loads and workloads by standardizing the flight operations. In addition, the flight trajectory can be easily predicted, and the air traffic can be efficiently managed.

The main objective of this study is to obtain a robust schedule in the PMS under uncertainties. To this end, two main problems are considered. The first problem is a scheduling problem without uncertainties. A holding pattern is implemented along with the PMS, and the effect of the scheduling algorithm with a holding pattern is analyzed. A holding pattern is usually defined near the PMS and used to manage the arrival time of a flight at the PMS. In congested scenarios, the PMS alone may fail in managing the arriving air traffic, requiring the air traffic controllers to manage the flights through radar vectoring, which

is laborious for the controllers. Nevertheless, by including the holding pattern in the scheduling problem of the PMS, the air traffic controllers can efficiently manage the flights even in saturated airspace. In addition, the degree of freedom of scheduling can be increased, thereby enhancing the quality of the scheduling solution.

The second problem is a robust scheduling problem considering uncertainties. In general, the presence of uncertainties can degrade the scheduling performance, causing the scheduling algorithm to yield an inappropriate flight schedule. Rescheduling must be performed to adjust the inappropriate schedule, thereby increasing the workload of the air traffic controllers. In addition, an additional delay of the flight may occur in the rescheduling process, leading to an increase in the overall delay of the airspace. Moreover, the freedom of scheduling for the PMS is limited compared to that of the radar vectoring because the operation of the flights is standardized. Owing to the lack of freedom of rescheduling, it is preferable to avoid excessive rescheduling in the PMS. Consequently, a robust schedule must be generated to address the uncertainties in the PMS. In this study, two types of uncertainties are considered, specifically, those pertaining to the Estimated Time of Arrival (ETA) and Continuous Descent Approach (CDA). The ETA determines the scheduling bounds of each flight at a specific point; the ETA is usually assumed to be deterministic but is stochastic in practical. Therefore, the ETA uncertainty must be considered to determine a feasible solution for a flight and minimize delays. By considering the uncertainties, the robustness-related quantities such as the amount of schedule change and number of constraint violations can be likely be reduced.

1.2 Literature Review

The literature review presented in this section is divided into three sections. At first, studies on the aircraft sequencing and scheduling is reviewed in Section 1.2.1. In Section 1.2.2, the literature related to the point merge system is summarized. Because a scheduling problem in the point merge system is rarely studied, the literature reporting the result of Human-in-the-loop simulation is also reviewed. Section 1.2.3 presents the review of studies on aircraft scheduling considering uncertainty.

1.2.1 Aircraft Sequencing and Scheduling

Studies on aircraft sequencing and scheduling have been mainly conducted focusing on the scheduling problem in a runway or the airport surface. The throughput of runway is the main concern of these studies.

In the beginning, only the arrival scheduling problem was studied. Beasley et al. proposed a population heuristic algorithm for London Heathrow airport, which showed that the heuristic algorithm can improve the runway throughput about 2-5%. [5] Lee et al. studied a trade-off relation between fuel cost, runway throughput, and delay cost. A dynamic programming algorithm was used to solve this optimization problem under the constraint of limited deviation from a first-come-first-served (FCFS) sequence. [6, 7] Hu et al. used a genetic algorithm in arrival scheduling, and the chromosome was represented as binary rather than permutation for efficiency. [8] Eun et al. presented an arrival sequencing and scheduling algorithm. The delay was modeled as discrete rather than continuous and computed by a genetic algorithm. [9] Ant colony system, a sort of meta-heuristic algorithm, was also used to solve the arrival schedul-

ing problem. [10] Harikopoulo et al. suggested a polynomial-time algorithm to calculate the optimal sequencing of the arrival aircraft. [11]

On the other hand, some studies concentrated on the departure problem. Rathinam et al. handled the departure scheduling problem using a chain model and a generalized dynamic programming. [12] Kim et al. proposed a robust gate assignment between arriving flights and departing flights to solve the gate conflict problem on departure metering. [13] Simaiakis et al. also concentrated on the departure problem and designed pushback rate control protocols, which predict the departure throughput and recommend a releasing rate from gates. Field tests showed that fuel use was reduced by an estimated 9 tons. [14, 15] Montoya et al. used a dynamic programming approach and minimized multiobjective performance index, including total aircraft delay and runway throughput. [16]

Later, the departure scheduling problem was extended to the departure and arrival scheduling problem. Hu et al. proposed a novel genetic algorithm for the departure and arrival scheduling to manage the airport capacity. [17] A receding horizon control technique was used with a genetic algorithm to handle the dynamic environment. Chen et al. suggested a multiple point scheduling scheme which can enable the integration of departure/arrival scheduling and runway assignment at the same time. [18] A sequential dynamic strategy was also studied to obtain a real-time solution and to take advantage of updated traffic information for the departure/arrival scheduling. [19] Chandrasekar et al. proposed a branch-and-bound algorithm to compute an optimal arrival and departure sequencing. [20] Cox et al. surveyed papers about the ground holding problem, a sort of scheduling problem, and compared by simulations with a static model and a dynamic model. [21] Khadilkar et al. addressed the problem

of aircraft delay and fuel consumption, and the airport surface was modeled by a network. [22]

Recently, the runway and airport surface scheduling algorithms have started to be extended to encompass the TMA. By considering rerouting airborne flights, studies on the scheduling in the TMA could improve the performance of the scheduling algorithm. Xue et al. handled the integration of departure and arrival problem considering the route structure of TMA with a nondominated sorting genetic algorithm. [23] Choi et al. studied the design of optimal route structure in the extended terminal airspace area using FCFS and mixed-integer linear programming (MILP). [24]

As presented above, in aircraft sequencing and scheduling, lots of methods have been adopted. Heuristic algorithms were presented at first, [5] and optimization based approaches such as dynamic programming [7, 12, 16] and MILP [18, 19, 24] were studied later. To conduct nonlinear and complex optimization, the population based optimization approach including genetic algorithm was also suggested. [8, 9, 17, 23] In this study, the MILP method of optimization-based approach is used, because it guarantees the optimality of the solution compared to other methods.

1.2.2 Studies on Point Merge System

The studies on the PMS have been focusing on the performance and effectiveness of the PMS. EUROCONTROL Experimental Centre proposed the PMS and proved that it has many advantages in terms of the workload of the air traffic controller, estimation of the flight trajectory, and fuel efficiency. [25, 26] Numerical and simple Human-in-the-loop (HITL) simulations were performed

to show that the PMS can be beneficial in terms of staffing, predictability, and the environment. A HITL simulation with human air traffic controllers was also performed to analyze the potential of the PMS. [4] The HITL simulation proved that the instructions of air traffic controllers were reduced by as much as 10% when using the PMS. Sahin et al. modeled the PMS and simulated it for Istanbul International Ataturk Airport, which has converging runways. [27] The simulation results were compared to the traditional vectoring, and showed that the total average number of instructions and the frequency occupancy was dramatically decreased for the PMS than for vectoring.

However, a scheduling problem in the PMS is rarely studied and few researchers have addressed the scheduling of the PMS. Liang et al. proposed a PMS framework and solved the scheduling problem with the simulated annealing method, which is one of the meta-heuristic algorithms. [28–30] Hong et al. suggested an optimal scheduling algorithm of the PMS with MILP and obtained a robust schedule under the uncertainty of CDA. [31] de Wilde studied the implementation of PMS in the TMA of Amsterdam Airport Schiphol and analyzed the advantages and drawbacks. [32]

Even though a holding pattern is operated together with the PMS, none of these studies directly consider the holding pattern. Therefore, in this study, a holding pattern is considered in the scheduling problem of the PMS.

1.2.3 Scheduling Under Uncertainty

Uncertainty received little attention in ATM research compared to the study on the scheduling problem without considering uncertainty. Two main approaches exist for the research on the scheduling considering uncertainty:

Deterministic approach and stochastic approach. [33, 34]

In a deterministic approach, which is also called robust optimization, the uncertainty is considered as the bound of uncertain variables and the robustness could be guaranteed through an additional buffer. Murça et al. presented a robust optimization method for aircraft departure scheduling under uncertainty in the taxi-out process. [35] The performance of the approach was validated by simulations based on the runway delay, time conformance, and runway throughput. Ng et al. proposed an efficient artificial bee colony algorithm to address the aircraft scheduling problem under the uncertainty of arrival and departure delay. To handle the uncertainty, the schedule of the aircraft was treated as not a specific time but a time window. [36] Hong et al. handled the uncertainty of CDA in the PMS by a deterministic approach. [31]

On the other hand, the stochastic approach addresses the uncertainty as a probability distribution, thus computation load may be heavier than the deterministic approach to evaluate the uncertainty. Solveling et al. studied a two-stage approach for runway operation under stochastic uncertainties in pushback delay, time spent on taxiway, and deviation from estimated arrival time. [37] Xue et al. considered the uncertainty of flight arrival times at waypoints and of departure times at runway in the integrated departure and arrival problem. [38] Taylor et al. adopted a genetic algorithm in air traffic flow management. [39,40] A Pareto front was generated by a multi-objective genetic algorithm, and the performance was analyzed based on the three performance indexes such as ground delay, en route delay, and arrival schedule delay. Bosson et al. suggested multistage stochastic optimization of airport surface operation considering the uncertainty of pushback delay and gate arrival delay. [41] The taxi time uncer-

tainty was also studied by Mori [42], where a tabu search algorithm, a sort of meta-heuristic method was used, to compute the departure time at the gate with the estimated taxi time information. Hong et al. also addressed the uncertainty of CDA in the PMS by a stochastic approach with a particle swarm optimization. [43]

Some studies concentrated on the sensitivity analysis of the uncertainty. Atkin et al. presented a decision support tool for runway scheduling, and investigated the effect of taxi time uncertainty via numerical simulations. [44] Lee et al. proposed two airport surface traffic optimization approaches and analyzed the degradation of the performance under uncertainty such as pushback times, runway exit times, taxi speeds, and runway separation times. [45]

The ETA uncertainty has a critical impact to generate a feasible schedule, but has received little attention so far. In addition, the uncertainty was not considered in the scheduling of the PMS except Ref. [31, 43]. Therefore, in this study, the ETA uncertainty and the CDA uncertainty are considered together, and the effect of each uncertainty is analyzed. Furthermore, the deterministic approach is used to exploit the latest information on the ETA in the scheduling.

1.3 Contributions

The main contributions of this study are summarized as follows.

Scheduling algorithm in the PMS including a holding pattern

A holding pattern is introduced into the scheduling algorithm of the PMS. A PMS scheduling algorithm without a holding pattern cannot manage flights in case the required airborne delay exceeds the maximum allowable delay in the PMS. In this problematic situation, human air traffic controllers should intervene and reschedule the flights using radar vectoring, which is the main factor of increasing workload for air traffic controllers. The proposed algorithm can handle this problem using a holding pattern without the intervention of air traffic controllers. Furthermore, a holding pattern provides an additional possibility to schedule flights to reduce airborne delays.

Robust scheduling algorithm in the PMS considering the uncertainties of ETA and CDA

A robust scheduling algorithm is proposed in the PMS, which considers the uncertainties of ETA and CDA. Because the uncertainty may degrade the performance of the scheduling algorithm, it is important to consider the uncertainty in the scheduling problem. In addition, the ETA determines the scheduling bounds of each flight at a specific point, and therefore the ETA uncertainty is critical to calculate a feasible solution for a flight and to minimize the undesirable delay. Different from CDA uncertainty, the probability distribution of the ETA uncertainty depends on the remaining flight time. In addition, the previous ETA uncertainty has a correlation with the next ETA uncertainty. In

this study, the ETA uncertainty is modeled by a multivariate Gaussian distribution based on the result of ETA estimation history data in previous research, in order to reflect the dependence on the remaining flight time and the correlation between time steps. The information of uncertainty distributions is used to generate the uncertainty for simulation and utilized in the robust scheduling algorithm.

Analysis in terms of practical quantities

The effects of ETA and CDA uncertainties are analyzed in terms of not only average delay but also the indices of robustness such as the amount of scheduling change and the number of constraint violations. Most scheduling algorithms have mainly focused on the total delay of flights at the goal. However, if the uncertainty exists, the optimal solution can be no longer an optimal solution. Moreover, to address the updated airspace situation, air traffic controllers reschedule the predetermined schedules. In this rescheduling process, the workload of air traffic controllers could increase according to the amount of schedule change, and the number of flights which violate the separation constraint. Therefore, the quality of the schedule is validated by the flight delay and the robustness under uncertainty.

1.4 Dissertation Outline

The organization of this study is as follows.

In Chapter 1, the background and motivation of this study are described, and related studies are given. Furthermore, the contributions of this study are presented.

In Chapter 2, the concept of the PMS is described and the airspace of interest is explained. Two problems considered in this study are summarized in this chapter.

In Chapter 3, the operation concept of the proposed scheduling algorithm is provided. The formulation of the normal scheduling algorithm ignoring uncertainty is explained, and the scheduling problem considering a holding pattern with the PMS is presented. The formulation of the robust scheduling algorithm is derived using the chance constraint of robust optimization. The uncertainty models of the ETA and the CDA are explained, and the way of uncertainty generation is presented. Finally, an explanation of the sliding window and the simulation procedure is summarized.

In Chapter 4, the performance of the scheduling algorithm with a holding pattern is verified by Monte Carlo simulations. The success rate and the average delay is compared to the FCFS algorithm and the scheduling algorithm without a holding pattern. In addition, the performance of the robust scheduling algorithm is demonstrated by Monte Carlo simulations. Overall 5 algorithms considering each uncertainty are simulated and show the effect of each uncertainty.

Chapter 5 summarizes the main results of this study and provides suggestions for future work.

Chapter 2

Problem Statements and Airspace of Interest

2.1 Airspace of Interest

2.1.1 Concept of Point Merge System

PMS is used to manage the flights in TMA and is composed of two main elements: Sequencing legs and merge point. Figure 2.1 shows the typical configuration of PMS. The sequencing leg is defined as an arc, thus the points on the sequencing leg have the same distance from the merge point.

An aircraft using the PMS passes the following process. At first, the aircraft enters the PMS through the initial point of the sequencing leg, and the aircraft flies along the sequencing leg until the CDA operation ("Direct to") is permitted by the air traffic controller. If the CDA operation is allowed, the aircraft descends to the merge point. After passing the merge point, the aircraft conduct a final approach and lands on the runway.

Because of the equidistance of the sequencing legs, the air traffic controller can easily change the arrival time of aircraft at the merge point by ordering "Direct to" action. In Fig. 2.1, Flight A conducted the CDA operation as soon as it entered the PMS, while Flight C flew to the end of the sequencing leg

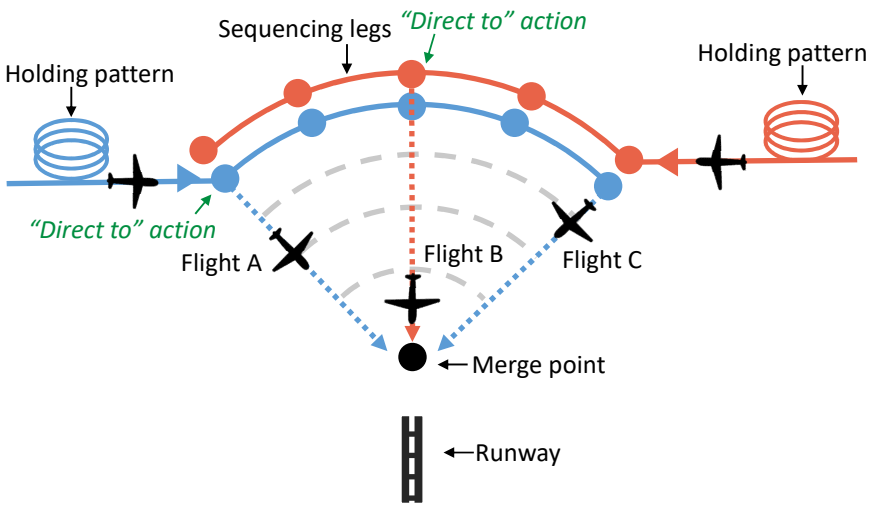


Figure 2.1: Typical configuration of PMS.

because it did not receive "Direct to" instruction. In addition, holding patterns are usually located before the entry point of the sequencing legs, and the air traffic controller can manage the arrival time of aircraft at the PMS.

PMS has several merits. First, the communications and workload of the air traffic controller could be reduced because the air traffic controller can manage aircraft through "Direct to" instruction. Also, it is easy to estimate the trajectory of aircraft, thereby the traffic controller and the pilot can recognize the airspace situation more clearly. Furthermore, fuel efficiency can be improved by the CDA operation. The CDA is well-known for its fuel efficiency by allowing each aircraft its optimal vertical trajectory, and airports implementing a PMS reported a significant amount of fuel-saving. [46, 47]

2.1.2 Holding Pattern

A holding pattern is a procedure charging additional delay for an aircraft in case of traffic congestion, poor weather, and other undesired situations. [48, 49] It is usually defined in aeronautical charts and composed of two semi-circles, two legs, and the direction reminding of a running track. Because the maximum speed and the time of straight flight are predefined in the aeronautical chart, the air traffic controller can manage the delay of a flight in a holding pattern easily by determining the number of laps for an aircraft.

Figure 2.2 shows the configuration of a holding pattern. If a flight is directed holding by an air traffic controller, the flight enters the holding pattern at the holding fix and flies along with the holding pattern as much as the air traffic controller commanded. Note that there are several holding patterns, which are separated by altitude. It is called a 'holding stack.' Assume that a flight is trying to enter the holding pattern, but another flight is already conducting the holding procedure. Then, the new flight should enter the upper holding pattern separated by the lower holding pattern to prevent a mid-air collision. When exiting the holding pattern, the flight in the lowest holding pattern exits the pattern first. Therefore, the first-in-first-out rule is valid in the holding stack.

As shown in Fig. 2.3, the holding pattern is used together with the PMS rather than used alone. In general, the holding pattern is located before the entry point of the PMS to manage the arrival time of the aircraft at the initial point of the PMS. However, it can be defined at other fixes such as the merge point of the end fix of the sequencing leg in the PMS.

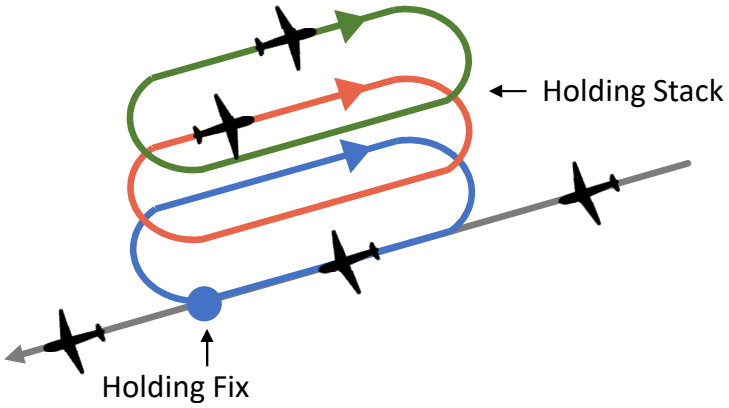


Figure 2.2: Configuration of holding stack.

2.1.3 Target Airspace

In this study, the PMS at Jeju International airport is mainly considered. The route between Jeju International Airport (CJU) and Gimpo International Airport (GMP) is known as one of the busiest routes in the world, of which annual flights are almost 80,000. [50] Whereas incredible flights use this route, only one runway can be affordable in CJU considering the wind direction. Therefore, it results in congestion around the airport and requires effective scheduling for better on-time performance.

Figure 2.3 shows the PMS of CJU in the direction of RWY 25. There are three inbound traffic flows in this PMS. The first route is PC731-DANBI-WOODO-HANUL routes in this chart (route 1), the second route is PC735-DANBI-WOODO-HANUL (route 2), and the third route is MAKET-SELIN-WOODO-HANUL (route 3). Based on the historical data, the percentage of flights using the PMS is 85.85% for route 1, 4.42% for route 2, and 9.73% for route 3. [31] In the PMS, two sequencing legs are defined. One is from DANBI

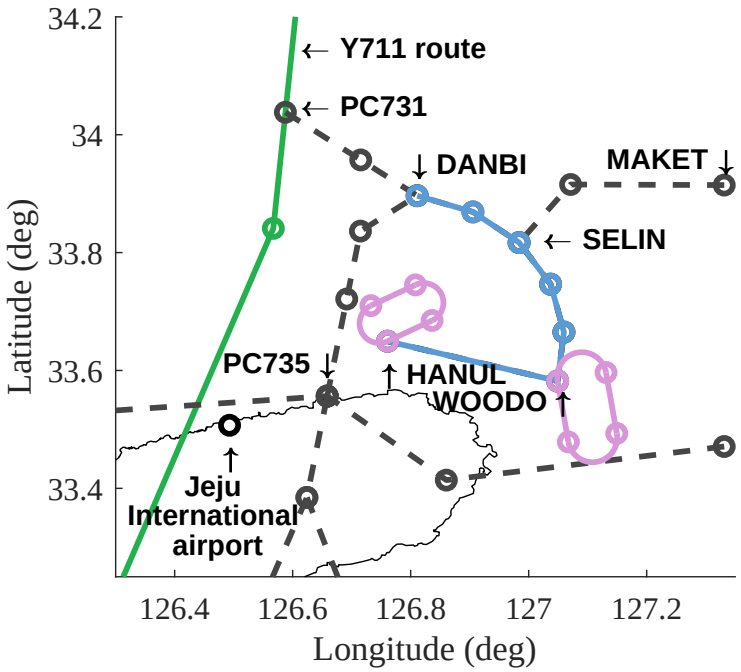


Figure 2.3: Standard terminal arrival route of Jeju international airport.

to WOODO and the other is SELIN to WOODO. Therefore, the flights of route 1 and route 2 enter the PMS through the first sequencing leg (DANBI), and the flights of route 3 enter the PMS through the second sequencing leg (SELIN). These two sequencing legs are separated vertically, which means that conflict does not occur between two sequencing legs. Note that two holding patterns exist at WOODO and HANUL to address the saturation of the PMS.

2.2 Problem Statements

In this study, two scheduling problems are considered. One is the scheduling problem without considering uncertainty, and the other is the scheduling problem considering uncertainty. In a scheduling problem, a flight maintains its scheduled time of arrival (STA) determined by the scheduling algorithm. However, the flight may not stick to the predetermined STA if uncertainty occurs. Therefore, rescheduling is inevitably required to generate a feasible schedule for a flight.

For the problem without considering uncertainty, scheduling is performed including a holding pattern. As shown in Fig. 2.3, holding patterns exist at WOODO and HANUL to manage the saturation of the TMA. Between two holding patterns, the holding pattern at WOODO is selected and used as an additional scheduling point. Therefore, the scheduling algorithm considering a holding pattern determines the arrival time at the entry point of the PMS, the time to enter the holding pattern, the time to exit the holding pattern, the time to start the CDA operation, and the arrival time at the merge point.

For the second problem considering uncertainty, scheduling is performed and rescheduling is also performed. With the ETA and the CDA time information of an aircraft, the scheduling algorithm determines the arrival time at the entry point of the PMS, the starting time to conduct the CDA operation, and the arrival time at the merge point. The information of the ETA and the CDA time include uncertainties inevitably, which may degrade the performance of the scheduling algorithm. Those uncertainties are modeled by Gaussian distribution and considered in the scheduling algorithm by the chance constraint. The ETA uncertainty, particularly, is modeled by the multivariate Gaussian random

variable to reflect the time-dependent characteristics. A detailed explanation of the scheduling algorithm and the model of the uncertainty is provided in Chap. 3 and Chap. 4.

Chapter 3

Scheduling Algorithm of Point Merge System

The scheduling algorithm proposed in this dissertation considers a holding pattern integrated with the PMS. The uncertainty is ignored in the algorithm, and therefore the rescheduling is unnecessary. The schedule of flights is determined by the scheduling algorithm, and the flight maintains the initial schedule for the whole simulation.

3.1 Scheduling Algorithm without Holding Pattern

A normal scheduling algorithm, which does not consider uncertainty and a holding pattern, has been widely studied. [31,51] Figure 3.1 shows the simplified node–link structure of the PMS of Jeju international airport. For convenience, the initial entry point is denoted as p_I, the turning point as p_T, and the merge point as p_F. The turning point is the conceptual point where the aircraft starts CDA operation, and therefore the turning point is on the sequencing leg. Let the route from DANBI to HANUL r_1 and the route from SELIN to HANUL r_2. Then, there exist two initial points, p_{I_1} and p_{I_2}, two turning points, p_{T_1} and p_{T_2}, and one merge point, p_F. Therefore, r_1 and r_2 mean (p_{I_1}, p_{T_1}, p_F) and (p_{I_2}, p_{T_2}, p_F), respectively. Note that two turning points are required because two

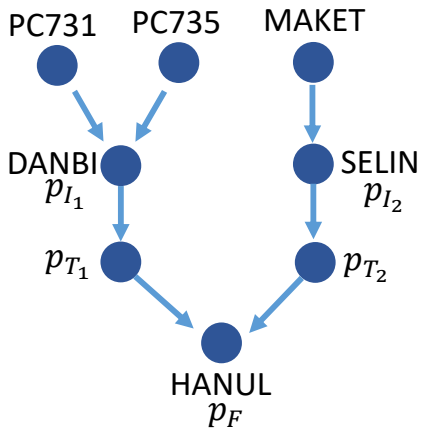


Figure 3.1: Node–link structure of the PMS in CJU.

sequencing legs are separated by altitude.

Several constraints, including the separation constraint and the single route constraint, are introduced in this study. Capozzi et al. represented the MILP framework of the traditional route structure [52], which is modified in this study to impose other important constraints.

Three main variables determine the scheduling result. First, $A_{f,r}$ is a binary variable that denotes whether flight f uses route r. It returns 1 if f uses r; otherwise, it returns 0. Second, $S_{f,f',r,r',p}$ decides the priority at point p, which is a common point of routes r and r', when flight f chooses route r and flight f' chooses route r'. If f is prior to f' at point p, then $S_{f,f',r,r',p}$ takes a value of 1; otherwise, it takes 0. Third, $T_{f,r,p}$ determines the arrival time of flight f at point p on route r. These three variables are used to formulate the MILP optimization problem.

In this study, the performance index to be optimized is defined as follows,

$$J = \sum_{f \in \mathbf{F}} \sum_{r \in \mathbf{R}} A_{f,r} T_{f,r,p_F} \tag{3.1}$$

where p_F is the last point on route r (merge point), and $\mathbf{F}$ and $\mathbf{R}$ denote the sets of flights and routes, respectively. The performance index is set as the sum of the transit time of all flights because the purpose of the proposed algorithm is to reduce airborne delay.

The following constraints are considered for optimal scheduling in the PMS.

Single Route Constraint:

$$\sum_{r \in \mathbf{R}} A_{f,r} = 1, \quad \forall f \in \mathbf{F} \tag{3.2}$$

This constraint is required because a flight can only select one single route among $\mathbf{R}$.

Ordering Constraint:

$$S_{f,f',r,r',p} + S_{f',f,r',r,p} = A_{f,r} A_{f',r'}, \qquad \forall f \neq f', r, r', p \in \mathbf{P}_r \bigcap \mathbf{P}_{r'} \tag{3.3}$$

where $\mathbf{P}_r$ is the set of points of route r. The ordering constraint is imposed to relate variables $S_{f,f',r,r',p}$ and $A_{f,r}$. If there exists a common point p between routes r and r', the priority should be determined by setting $(S_{f,f',r,r',p}, S_{f',f,r',r,p}) = (1,0)$ or $(S_{f,f',r,r',p}, S_{f',f,r',r,p}) = (0,1)$. Note that the common points between routes r_1 and r_2 can be described as $\mathbf{P}_{r_1} \bigcap \mathbf{P}_{r_2}$, and for this problem $\mathbf{P}_{r_1} \bigcap \mathbf{P}_{r_2} = \{p_F\}$.

Safe Separation Constraint:

$$\begin{aligned} S_{f,f',r,r',p}(A_{f',r'} T_{f',r',p} - A_{f,r} T_{f,r,p} - SEP_{f,f',p}) \geq 0, \\ \forall f \neq f', r, r', p \in \mathbf{P}_r \bigcap \mathbf{P}_{r'} \end{aligned} \tag{3.4}$$

If flight f' on route r' is prior to flight f on route r at point p, then the separation distance between two flights should be maintained. In this study, the safety separation rule according to the aircraft class defined by ICAO is used, [5, 53] which is summarized in Table 3.1. Safe separation in terms of the distance is converted into separation time considering the flight speed.

Transit Time Constraint:

$$A_{f,r}[T_{f,r,p_T} - T_{f,r,p_I}] \geq 0, \quad \forall f, r \tag{3.5a}$$

$$A_{f,r}[T_{f,r,p_T} - T_{f,r,p_I} - T_r^{leg,max}] \leq 0, \quad \forall f, r \tag{3.5b}$$

$$A_{f,r}[T_{f,r,p_F} - T_{f,r,p_T} - T^{CDA}] \geq 0, \quad \forall f, r \tag{3.5c}$$

This constraint is required to define the transit time between two points. Equation (3.5a) and Eq. (3.5b) constrain the transit time from p_I to p_T, and Eq. (3.5c) defines the time for the CDA operation.

Table 3.1: ICAO minimum separation standards according to the class of aircraft (NM).

Leading aircraft \ Trailing aircraft	Heavy	Large	Small
Heavy	4	5	6
Large	3	3	4
small	3	3	3

Initial Time Constraint:

$$T_{f,r,p_I} - T^E \geq 0, \quad \forall f, r \tag{3.6a}$$

$$T_{f,r,p_I} - T^L \leq 0, \quad \forall f, r \tag{3.6b}$$

where T^E is the earliest time at the initial point p_I, and T^L is the latest arrival time at the initial point p_I. In this study, T^E is set as the ETA minus 1 minute, and T^L is set as the ETA plus 3 minutes according to Lee et al. [7]

Because the formulation using the above variables is a nonlinear problem rather than a linear problem, additional variables are introduced to convert the nonlinear optimization problem into the mixed-inter linear programming as

$$\delta^T_{f,r,p} = A_{f,r} T_{f,r,p} \tag{3.7}$$

$$\delta^A_{f,f',r,r'} = A_{f,r} A_{f',r'} \tag{3.8}$$

where $\delta^T_{f,r,p}$ means an effective STA of flight f at point p on route r, and $\delta^A_{f,f',r,r'}$ returns 1 if flight f uses r and f' uses r'.

Then, the final form of the normal scheduling algorithm is defined as follows,

$$\text{Minimize} \quad J = \sum_{f \in \mathbf{F}} \sum_{r \in \mathbf{R}} \delta^T_{f,r,p_F} \tag{3.9}$$

subject to

$$\delta^T_{f,r,p_F} \leq M A_{f,r}, \quad \forall f, r \tag{3.10a}$$

$$\delta^T_{f,r,p_F} \leq T_{f,r,p_F} + M(1 - A_{f,r}), \quad \forall f, r \tag{3.10b}$$

$$\delta^T_{f,r,p_F} \geq T_{f,r,p_F} - M(1 - A_{f,r}), \quad \forall f, r \tag{3.10c}$$

$$\sum_{r \in \mathbf{R}} A_{f,r} = 1, \quad \forall f \in \mathbf{F} \tag{3.11}$$

$$-A_{f,r} + \delta^A_{f,f',r,r'} \leq 0, \quad \forall f \neq f', r, r' \tag{3.12a}$$

$$-A_{f',r'} + \delta^A_{f,f',r,r'} \leq 0, \quad \forall f \neq f', r, r' \tag{3.12b}$$

$$A_{f,r} + A_{f',r'} - \delta^A_{f,f',r,r'} \leq 1, \quad \forall f \neq f', r, r' \tag{3.12c}$$

$$\begin{aligned} S_{f,f',r,r',p} + S_{f',f,r',r,p} = \delta^A_{f,f',r,r'}, \\ \forall f \neq f', r, r', p \in \mathbf{P}_r \bigcap \mathbf{P}_{r'} \end{aligned} \tag{3.12d}$$

$$\begin{aligned} \delta^T_{f',r',p} - \delta^T_{f,r,p} - SEP_{f,f',p} + M(1 - S_{f,f',r,r',p}) \geq 0, \\ \forall f \neq f', r, r', p \in \mathbf{P}_r \bigcap \mathbf{P}_{r'} \end{aligned} \tag{3.13}$$

$$\delta^T_{f,r,p_T} - \delta^T_{f,r,p_I} \geq 0, \quad \forall f, r \tag{3.14a}$$

$$\delta^T_{f,r,p_T} - \delta^T_{f,r,p_I} - A_{f,r} T^{leg,max}_r \leq 0, \quad \forall f, r \tag{3.14b}$$

$$\delta^T_{f,r,p_F} - \delta^T_{f,r,p_T} - A_{f,r} T^{CDA} \geq 0, \quad \forall f, r \tag{3.14c}$$

$$\delta^T_{f,r,p_I} - A_{f,r}(ETA_f - T^E) \geq 0, \quad \forall f, r \tag{3.15a}$$

$$\delta^T_{f,r,p_I} - A_{f,r}(ETA_f + T^L) \leq 0, \quad \forall f, r \tag{3.15b}$$

where M is a large enough number, and ETA_f is a ETA of flight f.

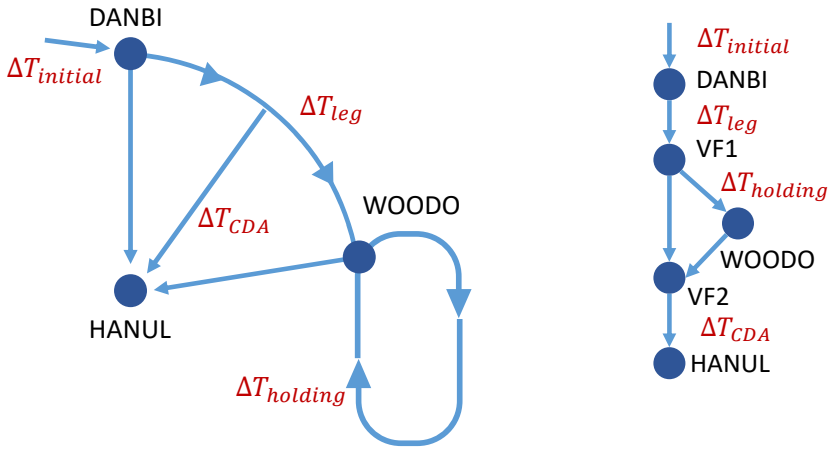
Note that the performance of the normal scheduling algorithm could be degraded in reality because it does not consider the uncertainty.

3.2 Scheduling Algorithm with Holding Pattern

As shown in Fig. 2.3, air traffic controllers often use a holding pattern at the end of the sequencing leg (WOODO fix) and at the merge point (HANUL fix) to address the saturation. In this study, only the holding pattern in the WOODO fix is considered to simplify the problem. Similarly, the incoming aircraft through the SELIN fix (MAKET-SELIN-WOODO-HANUL) are neglected because the percentage of the traffic through this route is less than 10%. Some airports prefer to locate the holding pattern at the initial point of the PMS (DANBI) than at the end of the sequencing leg. In that case, the location of the holding pattern can be easily adjusted by modifying some of the constraints.

Figure 3.2(a) shows the simplified version of the PMS with the holding pattern. The PMS with the holding pattern has several complicated conditions compared to the typical PMS, and therefore it is difficult to directly apply the MILP formulation to solve this problem. To address this problem, the original PMS structure is converted into the structure with two virtual fixes as shown in Figure 3.2(b). VF1 is the first virtual fix introduced in this study, which denotes the state when an aircraft finishes conducting the delay absorption. VF2 is required to assure a safe separation in the situation when an aircraft in the sequencing leg or holding pattern is ready to descend to the HANUL fix. The aircraft drops by the DANBI, VF1, WOODO, VF2, and HANUL fixes if the aircraft is scheduled to use the holding pattern. Otherwise, the aircraft only flies through the DANBI, VF1, VF2, and HANUL fixes. Detailed information about the routes and points is summarized in Table 3.2.

Compared to the normal scheduling algorithm without a holding pattern, the following constraints should be considered to include a holding pattern in



(a) Original structure.

(b) Transformed structure.

Figure 3.2: Node–link structure of the PMS in CJU with a holding pattern.

Table 3.2: Notation of fix points and routes.

Notation	Description
p_I	DANBI
p_{T_1}	VF1
p_H	WOODO
p_{T_2}	VF2
p_F	HANUL
r_1	$p_I \rightarrow p_{T_1} \rightarrow p_{T_2} \rightarrow p_F$
r_2	$p_I \rightarrow p_{T_1} \rightarrow p_H \rightarrow p_{T_2} \rightarrow p_F$

the scheduling problem.

Safe Separation Constraint:

$$S_{f,f',r,r',p}(A_{f',r'}T_{f',r',p} - A_{f,r}T_{f,r,p} - SEP_{f,f',p}) \geq 0, \quad \forall f \neq f', r, r', p \in \mathbf{P}_r \bigcap \mathbf{P}_{r'}, p \neq p_{T_1} \tag{3.16}$$

The separation constraint, Eq. (3.4), in section 3.1 should be replaced by Eq. (3.16). The difference between two constraints is whether the constraint is applied for point p_{T_1} or not. If the separation constraint is applied at point p_{T_1}, the unnecessary separation is imposed between aircraft entering a holding pattern and aircraft conducting the CDA. Although point p_{T_1} is excluded from the separation constraint, the separation distance between aircraft exiting a holding pattern and aircraft conducting the CDA can be guaranteed at point p_{T_2}. Therefore, the separation constraint should not be applied at point p_{T_1}.

Transit Time Constraint:

$$A_{f,r}[T_{f,r,p_{T_1}} - T_{f,r,p_I}] \geq 0, \quad \forall f, r \tag{3.17a}$$

$$A_{f,r}[T_{f,r,p_{T_1}} - T_{f,r,p_I} - T^{leg,max}] \leq 0, \quad \forall f, r \tag{3.17b}$$

$$A_{f,r}[T_{f,r,p_F} - T_{f,r,p_{T_2}} - T^{CDA}] \geq 0, \quad \forall f, r \tag{3.17c}$$

$$T_{f,r_1,p_{T_2}} - T_{f,r_1,p_{T_1}} = 0, \quad \forall f \tag{3.17d}$$

$$T_{f,r_2,p_{T_2}} - T_{f,r_2,p_H} = 0, \quad \forall f \tag{3.17e}$$

Compared to the normal scheduling without a holding pattern, Eqs. (3.17d)-(3.17e) are added to define the transit time between VF1, VF2, and WOODO. Because the transit time between VF1 and VF2 is 0 and the transit time between WOODO and VF2 is 0 in reality, Eqs. (3.17d)-(3.17e) are necessary. The transit time from VF1 to WOODO means the holding delay in a holding

pattern, thus it is defined in the holding constraint.

Holding Constraint:

$$A_{f,r_2}(T_{f,r_2,p_H} - T_{f,r_2,p_{T_1}} - m_f P) = 0, \quad \forall f \tag{3.18a}$$

$$T_{f,r_2,p_H} - T_{f,r_2,p_{T_1}} - \Delta T_{leg}^{max} = 0, \quad \forall f \tag{3.18b}$$

$$S_{f,f',r_2,r_2,p} = 1, \quad p \in \{p_{T_1}, p_H\}, \text{ if } f \text{ is prior to } f' \tag{3.18c}$$

where m_f is an integer variable between 1 and m_{max}, which denotes the number of holdings, and P is a parameter that indicates the time required of a single lap in a holding pattern. In the holding procedure, flights conduct discrete delay absorption, which results in different approaches to model a holding pattern, considering the discrete characteristics of the integer variable. Although P varies with the class of aircraft, it is assumed to be a constant value in this study. The situation where a flight enters the holding pattern implies that the flight has fully used the delay absorption in the sequencing leg, as illustrated in Eq. (3.18b). In addition, the First-In-First-Out (FIFO) rule is generally applied in the holding procedure. Thus, it is prohibited to overtake other aircraft in a holding pattern, and Eq. (3.18c) denotes this FIFO constraint.

Final Time Constraint:

$$T_{f,r,p_F} \leq T_1 \quad or \quad T_{f,r,p_F} \geq T_2, \quad \forall f, r \tag{3.19}$$

There may exist some situations that prohibit the landing procedure because of departing flights or bad weather conditions. In particular, because there is only one runway in CJU, the conflicts between the arriving and departing flights are significantly more severe at CJU than they are at other well-known airports. To address this situation, the final arrival time at the merge point should be

Table 3.3: Notation of auxiliary variables.

Notation	Formulation	Type
$\delta^T_{f,r,p}$	$A_{f,r}T_{f,r,p}$	continuous
$\delta^A_{f,f',r,r'}$	$A_{f,r}A_{f',r'}$	binary
$z_{f,r}$	-	binary
m_f	-	integer
δ^m_f	$A_{f,r}m_f$	continuous

determined to avoid the time between T_1 and T_2. The departure scheduling problem can be indirectly considered with this final time constraint. In addition, the effectiveness of scheduling with a holding pattern can be evaluated in a congested situation by this constraint.

The above constraints should be converted into MILP constraints with auxiliary variables and large M. Table 3.3 summarizes the introduced variables. The MILP formulations of the above constraints are as follows.

$$\begin{aligned}\delta^T_{f',r',p} - \delta^T_{f,r,p} - SEP_{f,f',p} + M(1 - S_{f,f',r,r',p}) \geq 0,\\ \forall f \neq f', r, r', p \in \mathbf{P}_r \bigcap \mathbf{P}_{r'}, p \neq p_{T_1}\end{aligned} \tag{3.20}$$

$$\delta^T_{f,r,p_{T_1}} - \delta^T_{f,r,p_I} \geq 0, \quad \forall f, r \tag{3.21a}$$

$$\delta^T_{f,r,p_{T_1}} - \delta^T_{f,r,p_I} - A_{f,r}T^{leg,max} \leq 0, \quad \forall f, r \tag{3.21b}$$

$$\delta^T_{f,r,p_F} - \delta^T_{f,r,p_{T_2}} - A_{f,r}T^{CDA} \geq 0, \quad \forall f, r \tag{3.21c}$$

$$\delta^T_{f,r_1,p_{T_2}} - \delta^T_{f,r_1,p_{T_1}} = 0, \quad \forall f \tag{3.21d}$$

$$\delta^T_{f,r_2,p_{T_2}} - \delta^T_{f,r_2,p_H} = 0, \quad \forall f \tag{3.21e}$$

$$\delta^T_{f,r_2,p_H} - \delta^T_{f,r_2,p_{T_1}} - \delta^m_f = 0, \quad \forall f \tag{3.22a}$$

$$\delta^m_f \leq M A_{f,r_2}, \quad \forall f \tag{3.22b}$$

$$\delta^m_f \leq m_f P + M(1 - A_{f,r_2}), \quad \forall f \tag{3.22c}$$

$$\delta^m_f \geq m_f P - M(1 - A_{f,r_2}), \quad \forall f \tag{3.22d}$$

$$S_{f,f',r_2,r_2,p} = 1, \quad p = p_2 \text{ or } p_3, \quad \text{for } f \text{ is prior to } f' \text{ in holding pattern} \tag{3.22e}$$

$$\delta^T_{f,r_2,p_H} - \delta^T_{f,r_2,p_{T_1}} - \Delta T^{max}_{leg} \leq \quad M(1 - A_{f,r_2}), \quad \forall f \tag{3.23a}$$

$$\delta^T_{f,r_2,p_H} - \delta^T_{f,r_2,p_{T_1}} - \Delta T^{max}_{leg} \geq -M(1 - A_{f,r_2}), \quad \forall f \tag{3.23b}$$

$$\delta^T_{f,r,p_F} - A_{f,r} T_1 \leq \quad M z_{f,r}, \quad \forall f, r \tag{3.24a}$$

$$\delta^T_{f,r,p_F} - A_{f,r} T_2 \geq -M(1 - z_{f,r}), \quad \forall f, r \tag{3.24b}$$

The final MILP formulations of scheduling algorithm with a holding pattern are Eqs. (3.9)-(3.12),(3.15), and Eqs. (3.20)-(3.24).

FCFS Algorithm

The First-Come-First-Served rule is the basic strategy for air traffic controllers. Therefore, in this study, the FCFS algorithm is considered to be scheduling by human air traffic controllers. The computation time of the FCFS algorithm is meaningless, and only the scheduling result is important. The FCFS

result can be obtained by adding the FCFS constraint to the scheduling algorithm and can be compared with that of the scheduling algorithm with a holding pattern. The FCFS constraint is as follows,

$$S_{f,f',r,r',p} = 1, \quad \text{for } f \text{ is prior to } f', \forall r, r', p \in \mathbf{P}_r \bigcap \mathbf{P}_{r'} \tag{3.25}$$

Chapter 4

Robust Scheduling Algorithm

4.1 Overview

The conceptual figure of the robust scheduling algorithm proposed in this dissertation is shown in Fig. 4.1. A sliding window is introduced because of the computation load and the lack of information. The scheduling is performed only for the flights within the sliding window. The initial scheduled time of arrival (STA) is assigned to the flight first entering the sliding window by the scheduling algorithm with the ETA and CDA time of the flight. Then, the previously determined STA is rescheduled by the scheduling algorithm. The ETA information is updated each time step, and therefore the constraint violation of ETA constraint may occur. The meaning of the violation of the ETA constraint is that the flight cannot stick to the previous STA by the change of ETA, thus the STA is not feasible. Therefore, the previous STA should be investigated to identify whether or not the STA conflicts with the ETA constraint, Eq. (3.15). If it occurs, the violation-free STA is calculated by the scheduling algorithm and assigned to the flight. If the constraint violation does not occur, then the flight maintains the previous schedule. This rescheduling is performed until the flight exits the PMS. In addition, the STA of the flight is fixed from a practical

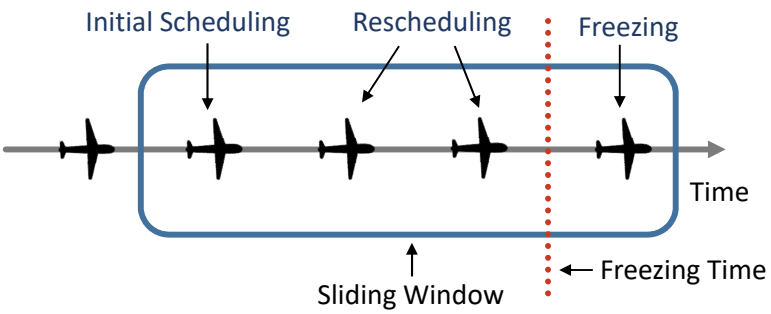


Figure 4.1: Conceptual figure of the robust scheduling algorithm.

point of view if the remaining flight time of the flight is less than the freezing time. [54]

4.2 Robust Scheduling Algorithm

In the normal scheduling algorithm, uncertainty is not considered. However, the uncertainty affects the quality of the scheduling, and therefore it is necessary to consider the uncertainty in a scheduling problem. In this study, two kinds of uncertainties are considered: ETA uncertainty and CDA uncertainty. Considering two uncertainties, some parameters in Eqs. (3.9)-(3.15) are converted from deterministic parameters to the parameters including uncertainty as follows,

$$\widetilde{ETA}_f = ETA_f + \Delta ETA_f \tag{4.1}$$

$$\widetilde{T}^{CDA} = T^{CDA} + \Delta T^{CDA} \tag{4.2}$$

where the tilde represents the parameter including the uncertainty, and Δ denotes the uncertainty of the parameter.

Now, the linear optimization problem of Eqs. (3.9)-(3.15) includes the estimated values of the parameters including the uncertainties. To deal with this problem, a robust optimization technique is adopted.

Let us consider a following general linear optimization problem.

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \end{aligned} \tag{4.3}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $x \in \mathbb{R}^n$.

Assume that one of the constraints of Eq. (4.3) has uncertain parameters ($\tilde{a}_{ij}$, $\tilde{b}_i$), and the parameters follow normal distributions as follows,

$$\sum_{l=1}^{n} \tilde{a}_{il} x_l \leq \tilde{b}_i \tag{4.4}$$

where $\tilde{a}_{ij} \sim N(a_{ij}, \sigma^2_{a,ij})$, and $\tilde{b}_i \sim N(b_i, \sigma^2_{b,i})$.

Because of the uncertain parameters, there exists a possibility that constraint (4.4) could be violated. Generally, it is not possible to completely eliminate the possibility of violation considering the characteristics of normal distribution. Therefore, it is desired that the probability of violation is restricted in a stochastic way. The stochastic constraint is called 'chance constraint,' [34] which is formulated as

$$P\{\sum_{l=1}^{n} \tilde{a}_{il} x_l > \tilde{b}_i\} \leq \kappa \tag{4.5}$$

where κ is a confidence level indicating the allowable maximum probability of the constraint violation.

It is assumed that the uncertain parameters follow normal distributions, and therefore the parameters can be expressed with standard normal random variable ($\xi \sim N(0, 1^2)$ as

$$\tilde{a}_{ij} = a_{ij} + \sigma_{a,ij}\xi_{ij} \tag{4.6a}$$

$$\tilde{b}_i = b_i + \sigma_{b,i}\xi_i \tag{4.6b}$$

Theorem 1 can be derived using the chance constraint, Eq. (4.5), and the characteristics of the normal distribution, Eq. (4.6).

Theorem 1. Given confidence level κ, the robust solution of Eq. (4.3) can be obtained by changing the uncertain constraint (4.4) with the following con-

straint.

$$\sum_{l=1}^{n} a_{il}x_l - b_i + \lambda\sqrt{\sum_{l=1}^{n} \sigma_{a,il}^2 x_l^2 + \sigma_{b,i}^2} \leq 0 \tag{4.7}$$

where λ satisfies $P\{\xi > \lambda\} = \kappa$, and $\xi \sim N(0, 1^2)$.

Proof. Substituting Eq. (4.6) into Eq. (4.5), we have the inequality probability condition of Eq. (4.5) as

$$\sum_{l=1}^{n} a_{il}x_l + \sum_{l=1}^{n} \sigma_{a,il}x_l\xi_{il} > b_i + \sigma_{b,i}\xi_i \tag{4.8a}$$

$$\sum_{l=1}^{n} \sigma_{a,il}x_l\xi_{il} - \sigma_{b,i}\xi_i > -\sum_{l=1}^{n} a_{il}x_l + b_i \tag{4.8b}$$

A linear combination of normal random variables still follows a normal distribution. Therefore, LHS of Eq. (4.8) follows a normal distribution with zero mean and variance $\sum_{l=1}^{n} \sigma_{a,il}^2 x_l^2 + \sigma_{b,i}^2$. If Eq. (4.7) is valid, then the chance constraint, Eq. (4.5), becomes

$$P\{\xi > \frac{-\sum_{l=1}^{n} a_{il}x_l + b_i}{\sqrt{\sum_{l=1}^{n} \sigma_{a,il}^2 x_l^2 + \sigma_{b,i}^2}}\} \leq P\{\xi > \lambda\} = \kappa. \tag{4.9}$$

□

Note that **Theorem 1** is an extended version of the previous result. [31]

The uncertain constraints of the normal scheduling algorithm are Eqs. (3.14c) and (3.15). Applying **Theorem 1** to Eqs. (3.14c) and (3.15), we have

$$\delta_{f,r,p_F}^T - \delta_{f,r,p_T}^T - A_{f,r}(T^{CDA} + \lambda_C\sigma_C) \geq 0, \quad \forall f, r \tag{4.10}$$

$$\delta_{f,r,p_I}^T - A_{f,r}(ETA_f + \lambda_E\sigma_{E,f} - T^E) \geq 0, \quad \forall f, r \tag{4.11a}$$

$$\delta^{T}_{f,r,p_I} - A_{f,r}(ETA_f - \lambda_E \sigma_{E,f} + T^L) \leq 0, \quad \forall f, r \tag{4.11b}$$

where subscript C and E denote CDA and ETA, respectively. As mentioned in Sec. 4.3, the uncertainty is assumed to follow normal distribution, i.e., $\widetilde{ETA}_f \sim N(ETA_f, \sigma^2_{E,f})$, $\widetilde{T}^{CDA} \sim N(T_{CDA}, \sigma^2_C)$. In summary, the formulation of the robust scheduling algorithm is Eqs. (3.9)-(3.14b), (4.10), and (4.11).

By comparing Eqs. (3.14c) and (3.15) with Eqs. (4.10) and (4.11), it can be found that the effective bounds of ETA and CDA are shrunk as much as $\lambda_E \sigma_{E,f}$ and $\lambda_C \sigma_C$. This additional buffer makes it possible to generate a robust solution.

In addition, Eq. (4.7) is a nonlinear inequality, and therefore it is impossible to use MILP formulation to obtain a robust solution in general. However, Eqs. (4.10) and (4.11) are still linear inequalities for the robust optimization problem, and MILP formulation can be used to obtain a robust scheduling solution.

4.3 Model of Uncertainty

In this study, two sorts of uncertainties are considered. Figure 4.2 shows those two uncertainties schematically. One is ETA uncertainty, ΔETA, which occurs before the aircraft arriving at p_I. The other uncertainty is CDA uncertainty, ΔT^{CDA}, which occurs during the CDA operation.

4.3.1 ETA uncertainty

The ETA uncertainty occurs while a flight approaches the initial point of the PMS. Because the ETA is estimated based on the data (aircraft category, wind information, position, velocity, etc.), the estimation error is inevitable. Several studies proved that the ETA error can be modeled as a normal distribution or a Johnson distribution. [55] From the perspective of the scheduling, the ETA error is considered as uncertainty because it is impossible to evaluate the ETA before the aircraft enters the PMS. Note that ETA error, i.e., ETA uncertainty, has important characteristics. Figure 4.3 shows the time-varying tendency of the ETA uncertainty. In Fig. 4.3, the time history of ETA error for 300 flights is shown based on the remaining flight time. [56] The ETA error decreases as an aircraft approaches the goal. In other words, the ETA uncertainty shows a time-varying probability distribution and the variance of it decreases as time goes by. This time-varying characteristics is applied in this study to reduce unnecessary buffer for flights. In addition, the ETA uncertainty of the current time step is closely related to the ETA uncertainty of the previous time step.

Therefore, in this study, a multivariate Gaussian distribution is used to consider the time-dependency and correlation of ETA uncertainty. The remaining flight time can be sampled at $t_1, t_2, \cdots, t_N$, and the ETA uncertainty

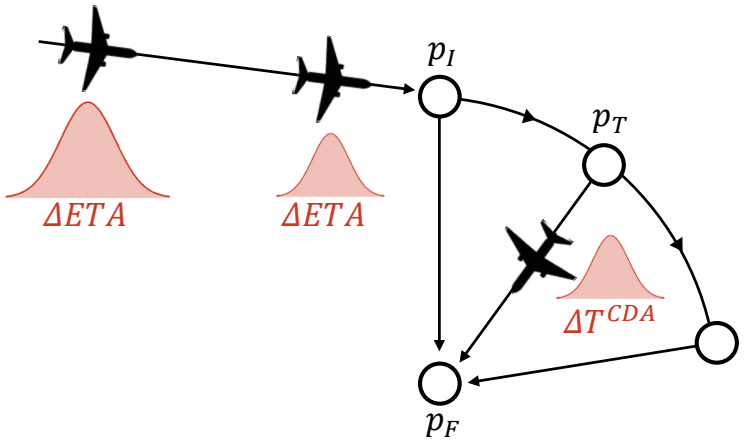


Figure 4.2: ETA and CDA uncertainty in the PMS.

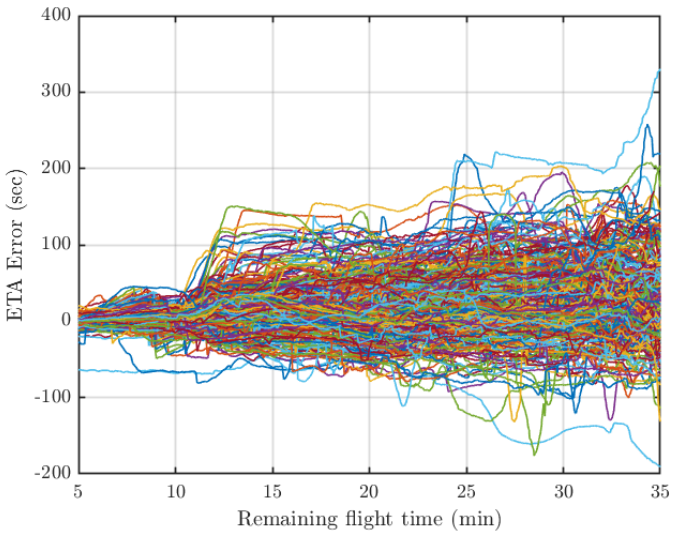


Figure 4.3: ETA error data with respect to the remaining flight time.

at the remaining flight time t_i can be represented as X_i. Then, the multivariate random variable of the ETA uncertainty, X, can be represented as $X = [X_1, X_2, \cdots, X_N]$. X follows a multivariate normal distribution $N(\mu, \Sigma)$ where $\mu \in \mathbb{R}^N$ and $\Sigma \in \mathbb{R}^{N \times N}$. Based on the data of Fig. 4.3, the average vector μ and the covariance matrix Σ can be obtained.

The ETA uncertainty for Monte-Carlo simulation can be generated by conditional distribution. The conditional distribution is computed as follows. Assume that $X_g = [X_{k+1}, X_{k+2}, \cdots, X_N]$ is given, and the objective is to calculate the probability distribution of $p(X_k|X_g) = p(X_k|X_{k+1}, X_{k+2}, \cdots, X_N)$. The marginal distribution $p(X_k, X_g)$ follows $N(\mu_{k:N}, \Sigma_{k:N})$, where $\mu_{k:N}$ is the subvector of μ and $\Sigma_{k:N}$ is the submatrix of Σ. Then, $\mu_{k:N}$ and $\Sigma_{k:N}$ can be partitioned as follows,

$$\mu_{k:N} = [\mu_k, \mu_g] \tag{4.12a}$$

$$\Sigma_{k:N} = \begin{bmatrix} \Sigma_{kk} & \Sigma_{kg} \\ \Sigma_{gk} & \Sigma_{gg} \end{bmatrix} \tag{4.12b}$$

Let us define $\Lambda = \Sigma_{k:N}^{-1}$, and Λ can also be partitioned as $\Sigma_{k:N}$.

$$\Lambda = \begin{bmatrix} \Lambda_{kk} & \Lambda_{kg} \\ \Lambda_{gk} & \Lambda_{gg} \end{bmatrix} \tag{4.13}$$

It is known that the conditional distribution, $p(X_k|X_g)$, follows the normal distribution $N(\mu_{k|g}, \Sigma_{k|g})$. [57]

$$\mu_{k|g} = \mu_k - \Lambda_{kk}^{-1}\Lambda_{kg}(X_g - \mu_g) \tag{4.14a}$$

$$\Sigma_{k|g} = \Lambda_{kk}^{-1} \tag{4.14b}$$

Therefore, the ETA uncertainty could be generated by Eq. (4.14). Using the above approach, the correlation of the ETA uncertainty between the previous time step and the current time step can be guaranteed. However, the possibility of collision due to the updated ETA still remains. If it occurs, the minimum separation is forcibly assigned between the flights during ETA updates.

The information on ETA distribution is required by the robust scheduling algorithm. In Eq. (4.11), σ_E is the standard deviation of ETA uncertainty. Note that the distribution of ETA uncertainty depends on the remaining flight time. Thus, the modeled ETA distribution, X, is summarized in Table 4.1 according to the remaining flight time (Time to go). The ETA uncertainty is modeled by a zero-mean normal distribution in the robust scheduling algorithm, but the ETA uncertainty in Table 4.1 has a nonzero (biased) average. Therefore, in this study, RMSE is used as a standard deviation of the ETA uncertainty rather than a standard deviation for simplicity.

4.3.2 CDA uncertainty

The other uncertainty considered in this study is CDA uncertainty, which occurs during the CDA operation. Compared to the ETA uncertainty, the CDA uncertainty does not show any significant time-varying tendency because the CDA operation is performed within 5 minutes. CDA uncertainty is known to follow a normal distribution, and therefore CDA uncertainty is generated based on the Gaussian distribution $N(0, 28^2)$. [31]

The robust scheduling algorithm also requires the information on CDA distribution. In Eq. (4.10), σ_C is the standard deviation of CDA uncertainty.

Table 4.1: ETA uncertainty model with respect to time to go.

Time to go(min)	Average(sec)	std(sec)	RMSE(sec)
5	0.74	6.56	6.59
10	1.65	12.83	12.92
15	19.50	34.92	39.95
20	18.41	40.94	44.83
25	28.46	49.66	57.16
30	29.54	52.17	59.88

4.4 Simulation Procedure with Scheduling Algorithm

Scheduling is usually performed at every time step. However, for each time step, it is not possible to consider every flight because of the computation load and the lack of suitable information. The computation load is closely related to the number of flights, and therefore a sliding window or receding horizon concept is adopted in this study. [58] In the sliding window technique, the scheduling is conducted only for the flights within the sliding window. As time step increases, a flight entering the sliding window is considered in the scheduling and a flight getting out of the sliding window is excluded in the scheduling.

In this study, the simulation process is conducted as shown in Fig. 4.4. At the first time step, the sliding window is updated and the scheduling is performed for the flights in the sliding window. The normal scheduling algorithm or the robust scheduling algorithm can be used as a scheduling algorithm. At the next time step, the sliding window and uncertainty are updated. ETA uncertainty is updated only for the flights approaching the PMS, and the ETA is fixed for the flights already entered the PMS. CDA uncertainty is injected only for the flights performing the CDA operation. Then, a rescheduling constraint is added for a scheduling algorithm. The newly added flights in the sliding window are scheduled by the original scheduling algorithm. However, flights already in the sliding window need to be slightly rescheduled according to the updated uncertainties, because some constraints might be violated by the updated ETA and CDA. To restore the constraints, an additional delay is necessary for the violating flights and the subsequent flights. Then, scheduling of the flights in the sliding window is performed by a scheduling algorithm with the rescheduling constraint. The freezing constraint is also considered in rescheduling as mentioned in 4.1. This

process is repeated for the whole time steps.

Algorithm 1 summarizes the simulation procedure of the scheduling algorithm. Here, Ω_k denotes the set of flights in sliding window at the k-th time step, and the superscript k of variable $\delta^{T,k}_{f,r,p_F}$ means the time step of scheduling variable. Note that there is a penalty for the flights violating constraints. Comparing Eq. (3.15a) and the rescheduling constraint in **Algorithm 1**, the rescheduling constraint has a penalty as much as T^E. It is because i) it is not desirable to change the previous schedule, and ii) it is almost impossible for a flight to accelerate as the flight approaches the goal point from a practical point of view.

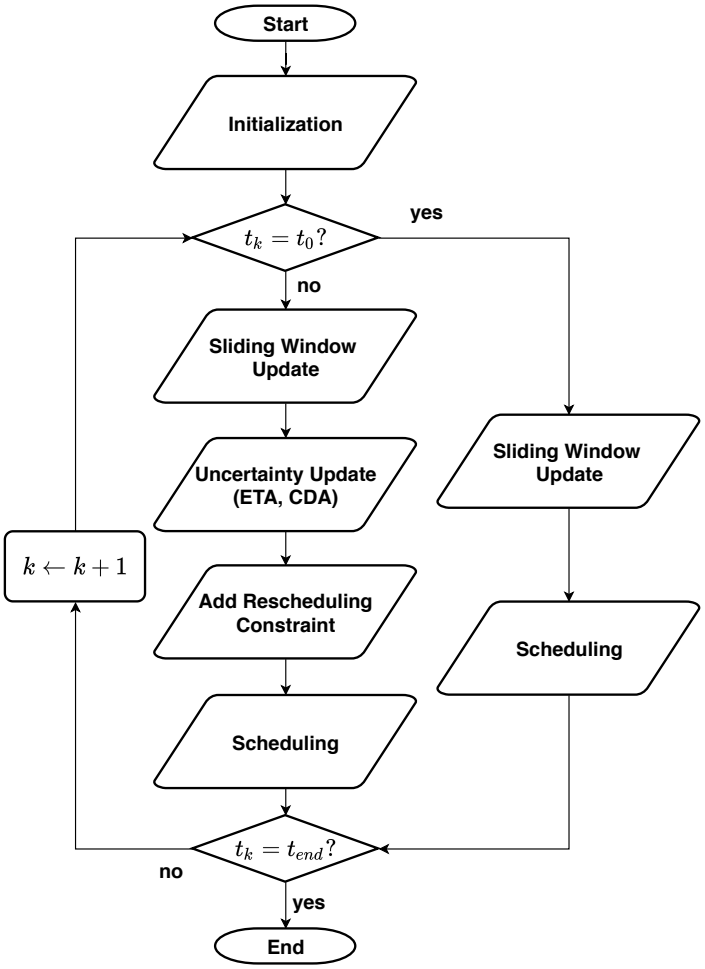


Figure 4.4: Flowchart of simulation procedure.

Algorithm 1 Simulation procedure of scheduling algorithm.

Require: ETA_f, aircraft category

initialization : $\Delta ETA_f \leftarrow 0$, $\Delta T^{CDA} \leftarrow 0$, $\Omega_0 \leftarrow \{f_1, f_2, \cdots, f_N\}$

for $t_k \leftarrow t_1$ **to** t_{end} **do**

  if $t_k = t_1$ **then**

    for $f \leftarrow f_1$ **to** f_N **do** ▷ Sliding window update

      $\Omega_k \leftarrow \{f\} \cup \Omega_{k-1}$ if $ETA_f \in [t_k, t_k + T_W]$,

        otherwise $\Omega_k \leftarrow \Omega_{k-1} - \{f\}$

    end for

    Schedule $\forall f \in \Omega_k$ with scheduling algorithm ▷ Scheduling

    return $\delta^{T,k}_{f,r,p}$

  else if $t_k > t_1$ **then**

    for $f \leftarrow f_1$ **to** f_N **do** ▷ Sliding window update

      $\Omega_k \leftarrow \{f\} \cup \Omega_{k-1}$ if $ETA_f \in [t_k, t_k + T_W]$ and $\delta^{T,k-1}_{f,r,p_F} \geq t_k$,

        otherwise $\Omega_k \leftarrow \Omega_{k-1} - \{f\}$

    end for

    for $f \leftarrow f \in \Omega_k$ **do**

      Generate ΔETA_f if $ETA_f > t_k$, otherwise $\Delta ETA_f \leftarrow 0$

      $ETA_f \leftarrow ETA_f + \Delta ETA_f$ ▷ ETA uncertainty update

      Generate ΔT^{CDA} if $\delta^{T,k-1}_{f,r,p_T} \leq t_k \leq \delta^{T,k-1}_{f,r,p_F}$ and

        $\delta^{T,k-1}_{f,r,p_F} - \delta^{T,k-1}_{f,r,p_T} < T^{CDA} + \Delta T^{CDA}$,

        otherwise $\Delta T^{CDA} \leftarrow 0$

      $T^{CDA} \leftarrow T^{CDA} + \Delta T^{CDA}$ ▷ CDA uncertainty update

    end for

23: **for** $f \leftarrow f \in \Omega_k \cap \Omega_{k-1}$ **do** ▷ Rescheduling Constraints

24: $\forall r$, add constraint $\delta^{T,k}_{f,r,p_I} \geq A_{f,r} ETA_f$ if $ETA_f - T^E \geq \delta^{T,k-1}_{f,r,p_I}$

25: or $ETA_f + T^L \leq \delta^{T,k-1}_{f,r,p_I}$

26: $\forall r$, add constraint $\delta^{T,k}_{f,r,p} \geq \delta^{T,k-1}_{f,r,p}$

27: $\forall r$, add constraint $\delta^{T,k}_{f,r,p_I} = \delta^{T,k-1}_{f,r,p_I}$ if $t_k \geq \delta^{T,k-1}_{f,r,p_I} + T^F$

28: **end for**

29: **Schedule** $\forall f \in \Omega_k$ with scheduling algorithm ▷ Scheduling

30: **return** $\delta^{T,k}_{f,r,p}$

31: **end if**

32: **end for**

Chapter 5

Numerical Simulations

Two problems are considered in the numerical simulation. Section 5.1 presents the simulations results of the scheduling problem in the PMS with a holding pattern. The uncertainty is ignored, and therefore the initial schedule determined by a scheduling algorithm is sufficient and the rescheduling process is unnecessary. In Section 5.2 the simulation results of the scheduling algorithm in the PMS are provided considering the uncertainties of the ETA and CDA. The rescheduling process is conducted to address the uncertainties, and the sliding window concept is adopted to reduce the computation load.

5.1 Simulation with Holding Pattern

A numerical simulation is performed to demonstrate the performance of the scheduling algorithm in the PMS with a holding pattern (PMS holding), described in Chapter 3. Two specific scenarios are selected to show the scheduling result of the PMS holding (Section 3.2), PMS (Section 3.1), and FCFS algorithms (Section 3.2). A Monte Carlo simulation is conducted to compare the three algorithms, and their performances are analyzed in terms of delay, computation time, and other factors.

There are three types of delay, as shown in Fig. 3.2: i) $\Delta T_{initial}$, the delay

before entering the PMS via speed control, ii) ΔT_{leg}, the time spent on the sequencing leg, and iii) $\Delta T_{holding}$, the discrete time delay in the holding pattern. These delays are used as control variables to schedule the flights. In this study, $\Delta T_{initial}$ is set between -1 minute and 3 minutes. In the standard terminal arrival route procedure (Figure 2.3), the speed of flight in the sequencing leg is 220 kts, and the length of the leg is 25 NM. Thus, the maximum of ΔT_{leg} is calculated as 409 seconds. For the holding procedure, the maximum number of laps is 3, because an overly long delay should be absorbed not by a holding pattern but by radar vectoring. P, i.e., the time required to complete one lap, is set to 240 seconds (4 minutes), which implies that the maximum holding time for an aircraft is 720 seconds. The CDA operation time, ΔT_{CDA}, is assumed to be a constant and set to 245 seconds. The separation time for safety between aircraft is determined based on the ICAO standard in Table 3.1, and 60 seconds is added to the separation time as a safety buffer to handle the uncertainty.

All simulations are conducted with desktop PC (I7-7700 Intel Core processor and 32 GB memories). MATLAB is selected as the simulation software, and CPLEX is used to solve the MILP problem. [59, 60] CPLEX is a commercial optimization software and is considered as a standard tool for solving integer programing problems.

5.1.1 Illustrative Simulation Result

The PMS holding algorithm can address wide-range scenarios, which the PMS algorithm cannot. The PMS holding algorithm has a much longer maximum allowable delay per flight than does the PMS algorithm, because the former includes a holding pattern. Therefore, for a fair comparison, the simu-

lation results are analyzed separately, i.e., when the PMS algorithm succeeds and when the PMS algorithm fails.

Figures 5.1-5.3 show the scheduling result of the three algorithms when the PMS algorithm succeeds. Eleven horizontal rectangles denote the time durations in the PMS of all aircraft. The left side of the rectangle is the initial time of an aircraft at the initial point, and the right side of the rectangle is the final time of an aircraft at the merge point. The rectangle is divided into three colored regions. The blue one indicates ΔT_{leg}, the time duration in the sequencing leg of each aircraft. The green represents $\Delta T_{holding}$, the time duration in a holding pattern. The gray one implies ΔT_{CDA}, the required time to perform the CDA operation. The 'x' mark indicates the ETA of each aircraft at the initial point. Thus, by comparing the 'x' mark and the left edge of the rectangle, $\Delta T_{initial}$ can be determined, whether an aircraft arrived at the initial point early. A shaded zone denotes the avoidable arrival time at the merge point, as mentioned in equation (3.19). During this time, all aircraft are prohibited from arriving at the merge point.

In Figure 5.1, aircraft 5 (AC5) uses a holding pattern once to lengthen the time duration in the PMS of the aircraft and avoid the shaded zone. Aircraft 1 (AC1), on the other hand, descends to the merge point immediately after it enters the PMS because the delay is unnecessary. The aircraft category of each aircraft is also described as a capital letter (H, L, S) next to the aircraft number on the y-axis. To make it easier to understand the simulation result, the flight trajectories in Figs. 5.1-5.3 are shown in Figs. 5.4-5.6.[1] Because the difference among three algorithms come from the sequence of AC5-AC11, only

[1]The visualized video is available at `https://youtu.be/t6h5I1leDdA`.

the snapshots after the prohibited time are shown. At $t = 1,520$ sec., the merge point is blocked with a red circle denoting the prohibited time. The red flights means that the flight used the route 1 and the blue flights come from the route 2. Aircraft 5 (AC5) is represented with yellow because it only use a holding pattern in the PMS holding algorithm. In Fig. 5.4, the sequence of AC5-AC11 is (5-6-7-8-9-10-11). On the other hand, the sequence of AC5-AC11 is (7-5-6-8-9-10-11) by swapping the sequence of AC5 and AC7. Compared to the PMS algorithm, the sequence of AC5-AC8 is changed by the holding pattern and the final sequence is (7-8-6-5-9-10-11). Note that the sequence change occurs at the sequencing leg and the holding pattern.

Figures 5.1-5.3 show that the PMS holding algorithm and PMS algorithm change the sequence of aircraft at the merge point to reduce airborne delay, whereas the FCFS algorithm maintains its original sequence. Table 5.1 demonstrates this tendency of algorithms to reduce delay. The PMS holding algorithm shows the best performance in terms of average delay (delayed time per aircraft), whereas the FCFS algorithm presents the worst performance. Because the FCFS algorithm is considered as a strategy of human air traffic controllers in this study, the PMS holding and PMS algorithms are expected to reduce the delay of aircraft compared to the current scheduling strategy in practice. It is interesting to know that the PMS holding algorithm provides better performance in delay than the PMS algorithm does, because the main purpose of the PMS holding algorithm is to expand the region of scenarios, which the PMS algorithm cannot schedule because of the limitation of maximum delay. The introduction of a holding pattern appears to provide another degree of freedom in the swapping sequence. Meanwhile, the PMS holding algorithm shows the

worst performance in computation time, because the extension of the PMS algorithm such as introducing integer variables requires more computation load than the PMS algorithm does. However, the computation time of the PMS holding algorithm remains acceptable and less than 10 seconds.

Figures 5.7 and 5.8 show the scheduling result of the three algorithms when the PMS algorithm fails. Because the required delay of an aircraft exceeds the maximum allowable delay of the PMS algorithm, the PMS algorithm fails to schedule. Therefore, only the results of the PMS holding and FCFS algorithms are shown in Figs. 5.7-5.8 and are summarized in Table 5.2. Similar to the previous simulation, the PMS holding algorithm swaps the sequence of some aircraft to minimize the average delay. The average delay of the PMS holding algorithm decreases by 6.52% compared to that of the FCFS algorithm. However, the computation time is much longer. The computation time of the FCFS algorithm almost doubles, whereas that of the PMS holding algorithm increases by approximately 4 times. Thus, the PMS holding algorithm is more vulnerable to an increase in computation load than the other two algorithms.

Note that some aircraft may sacrifice their delays to minimize the average delay, because minimizing the average delay and minimizing the maximum delay provide different solutions. Therefore, the fairness between aircraft can be a practical issue. In this study, this fairness issue is handled indirectly by constraining the maximum delay with the number of holding and the length of the sequencing leg. A bi-objective optimization could be used to directly address the fairness problem. [61]

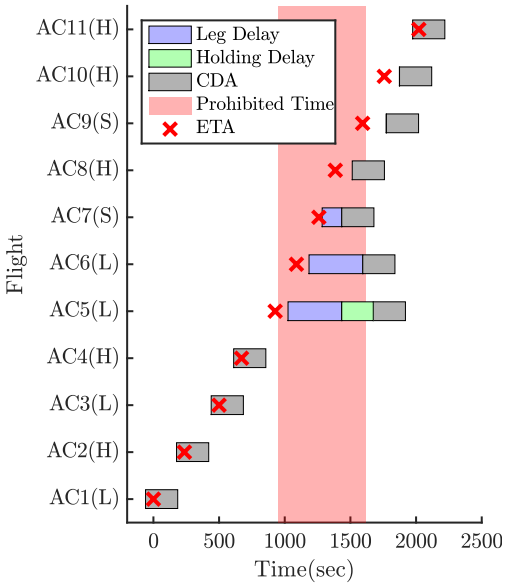


Figure 5.1: Scheduling result of PMS holding algorithm. (PMS algorithm succeeds.)

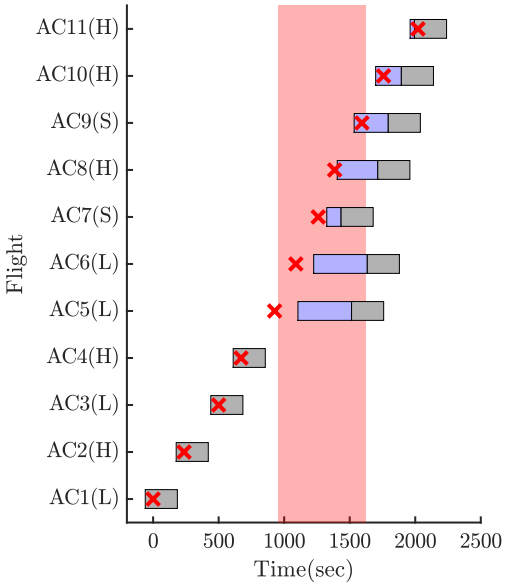


Figure 5.2: Scheduling result of PMS algorithm. (PMS algorithm succeeds.)

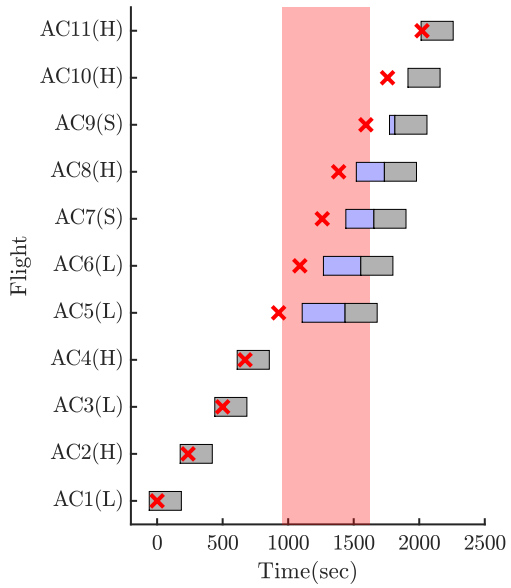
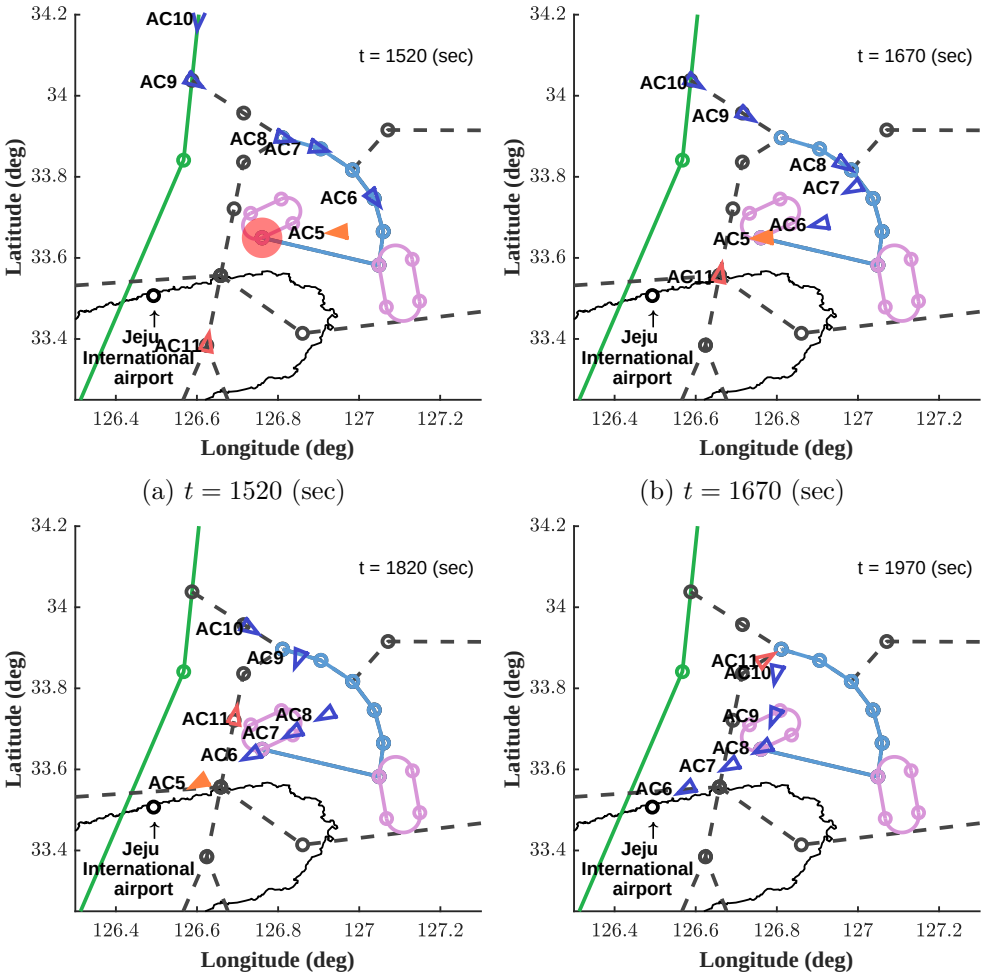


Figure 5.3: Scheduling result of FCFS algorithm. (PMS algorithm succeeds.)

Table 5.1: Scheduling result when the PMS algorithm succeeds.

Result(sec)	PMS Holding	PMS	FCFS
Average delay	202.27	215.00	227.73
Average interval	203.4	205.4	207.4
Computation time	0.6287	0.0206	0.2718



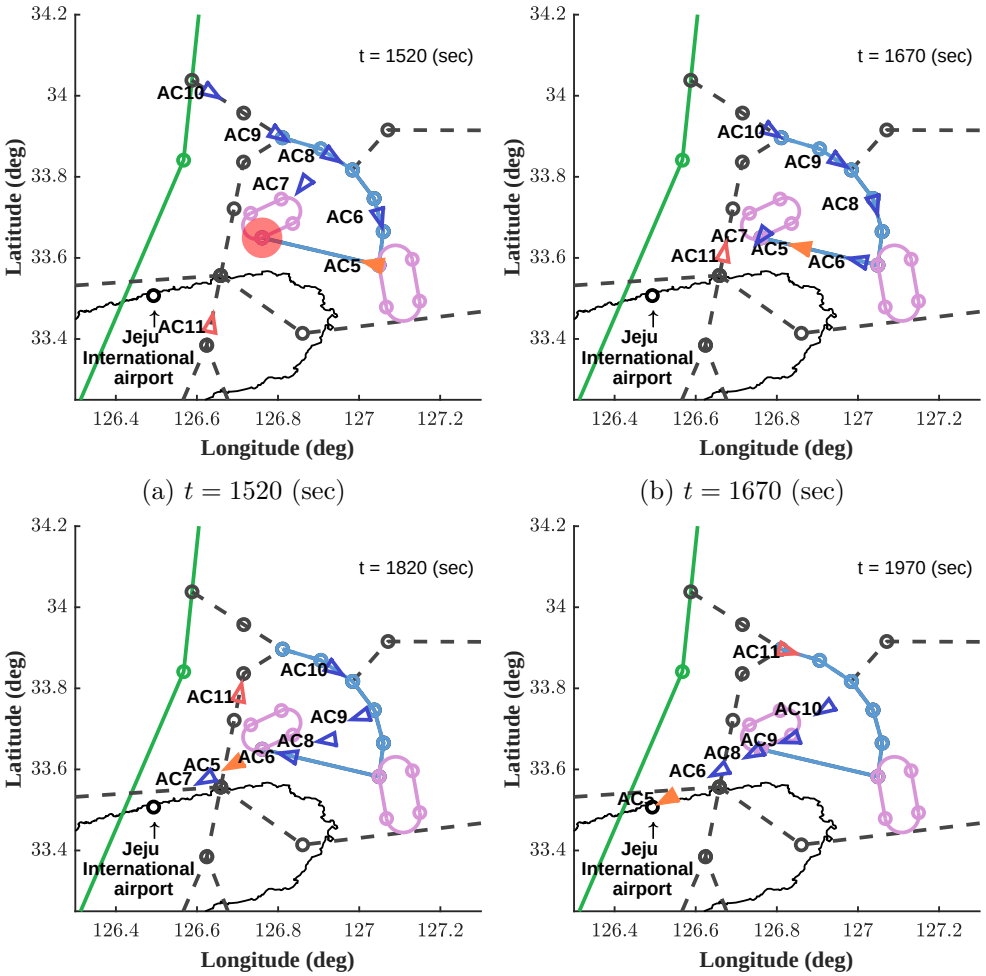
(a) $t = 1520$ (sec)

(b) $t = 1670$ (sec)

(c) $t = 1820$ (sec)

(d) $t = 1970$ (sec)

Figure 5.4: Flight trajectories of illustrate example. (FCFS)



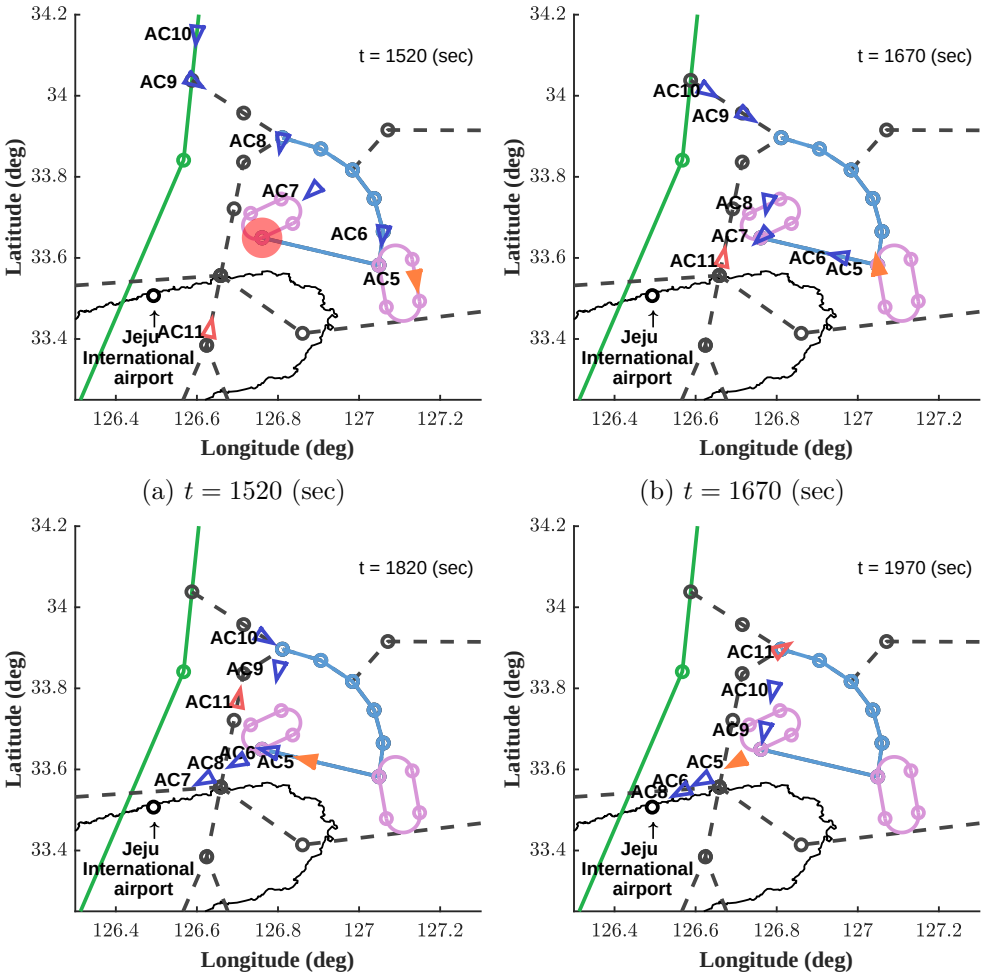
(a) $t = 1520$ (sec)

(b) $t = 1670$ (sec)

(c) $t = 1820$ (sec)

(d) $t = 1970$ (sec)

Figure 5.5: Flight trajectories of illustrate example. (PMS)



(a) $t = 1520$ (sec)

(b) $t = 1670$ (sec)

(c) $t = 1820$ (sec)

(d) $t = 1970$ (sec)

Figure 5.6: Flight trajectories of illustrate example. (PMS holding)

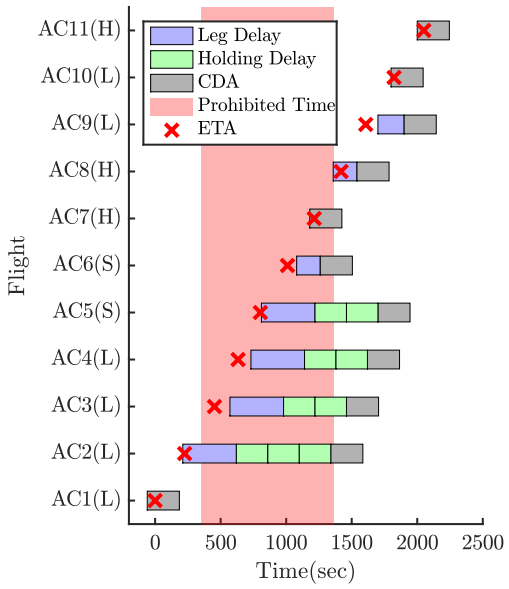


Figure 5.7: Scheduling result of PMS holding algorithm. (PMS algorithm fails.)

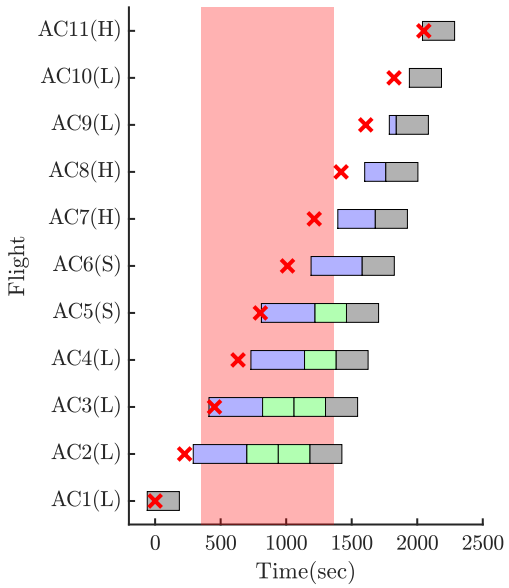


Figure 5.8: Scheduling result of FCFS algorithm. (PMS algorithm fails.)

Table 5.2: Scheduling result when the PMS algorithm fails.

Result(sec)	PMS Holding	FCFS
Average delay	469.36	502.09
Average interval	206	210
Computation time	2.8111	0.1302

5.1.2 Monte Carlo Simulation Result

Monte Carlo simulations are conducted to validate the performance of the PMS holding algorithm. To mimic a realistic situation, three aircraft densities (High, Medium, and Low) are considered based on the flight data analysis of Jeju International airport. The scenario time horizon assumed is set 30 minutes. Within 30 minutes, 11 aircraft enter the PMS for the high density; 7 aircraft for the medium density, and 5 aircraft for the low density. For each density, 100 scenarios are generated and used to perform the Monte Carlo simulation.

Several random variables are selected for Monte Carlo simulations. The ETA of each flight is assigned uniformly randomly and adjusted to satisfy the separation constraint at the initial point. The category of each aircraft is chosen based on the following probabilistic distribution: 40% for the heavy class, 50% for the large class, and 10% for the small class. The prohibited landing time $[T_1, T_2]$, defined in Eq. 3.19, is also randomly selected; T_1 is determined uniformly randomly and T_2 is chosen uniformly at random to make $T_2 - T_1$ between 1 and 20 minutes. The random variables used in the Monte Carlo simulations are summarized in Table 5.3.

As presented in Section 5.1.1, the PMS algorithm may fail at scheduling because of the limitation of maximum delay. If a failure occurs, a human air traffic controller should use radar vectoring to find proper solution. However, the radar vectoring approach requires more workload than the PMS, and therefore it is not desirable from a practical point of view. Table 5.4 represents the ratio of intervention by an air traffic controller among 100 Monte Carlo simulations for each density. The PMS holding and FCFS algorithms succeed in all scenarios, whereas the PMS algorithm fails to schedule approximately 30 scenarios for each

density. This result proves that the PMS holding algorithm expands the region of scheduling-possible scenarios by introducing a holding pattern compared to the PMS algorithm.

Among the 100 high-density scenarios, 62 scenarios can be scheduled by the PMS algorithm; in 20 scenarios, the delay is reduced compared with the result obtained by the FCFS algorithm. Thus, approximately 32.2% of the 62 scenarios shows a tendency of delay reduction. Selected results in which the delay is reduced by the PMS holding and PMS algorithms are shown in Figure 5.9. The amount of reduced average delay and the percent of reduced average delay are calculated based on the result of the FCFS algorithm. It can be found that 8 of the 20 delay-reduced scenarios show better performance in the PMS holding algorithm (solid line) than in the PMS algorithm (dashed line). This delay reduction performance is also observed in Table 5.5. The PMS algorithm shows a shorter average delay than the FCFS algorithm does, and the PMS holding algorithm shows a significantly shorter average delay than the PMS algorithm does. The average interval between aircraft, which is closely related to the throughput of the runway, also provides a similar tendency to the average delay, although the computation time indicates an opposite trend. In addition, the gap between the algorithms evidently decreases when the aircraft density decreases, because there is a positive correlation between the possibility of the swapping sequence and aircraft density.

The result when the PMS algorithm fails to schedule is summarized in Figure 5.10 and Table 5.6. Among the 38 scheduling-failed scenarios, 21 scenarios present a reduction of delay compared to the FCFS algorithm, for a percentage of approximately 55%. Note that the delay reduction percent is 32.2% for the

PMS success case. In addition, by comparing Tables 5.5 and 5.6, it can be found that the reduced amount of average delay by FCFS is bigger in PMS-failed scenarios than in PMS-succeeded scenarios. The computation time, however, increases for the PMS-failed case. As an example, for high density, the computation time that PMS succeeds is 0.2914 sec, but the computation time that PMS fails is 0.9908 sec. By contrast, there is no clear difference in the computation times of the FCFS algorithm. Therefore, it can be concluded that the PMS holding algorithm is likely to be affected by both the computation load and the problem size.

The above analyses of the PMS holding algorithm reveal that the PMS holding algorithm can schedule more severe scenarios by using a holding pattern than can the PMS algorithm alone. However, at the same time, the PMS holding algorithm is adversely affected by the computation time and is vulnerable to both the computation load and the problem size. Thus, it is important to select the appropriate parameters for scheduling. Table 5.7 illustrates the effect of m_{max} on the computation time of the PMS holding algorithm. As m_{max} increases, the intervention rate decreases. And every scenario can be scheduled by the PMS holding algorithm when $m_{max} = 3$. In terms of the intervention rate, it is desirable to increase m_{max} as much as possible, but the computation time increases as m_{max} increases, which is undesirable. Therefore, m_{max} should be properly selected while considering the trade-off between the intervention rate and the computation time.

Table 5.3: Random variables in Monte Carlo simulation.

Random Variables	Probability Distribution
ETA_f	uniform between 0 and 30 min.
Category	Heavy : 40% Large : 50% Small : 10%
T_1	uniform between 0 and 30 min.
$T_2 - T_1$	uniform between 1 and 20 min.

Table 5.4: Intervention rate of algorithms (%).

Density	PMS Holding	PMS	FCFS
High	0	38	0
Medium	0	34	0
Low	0	30	0

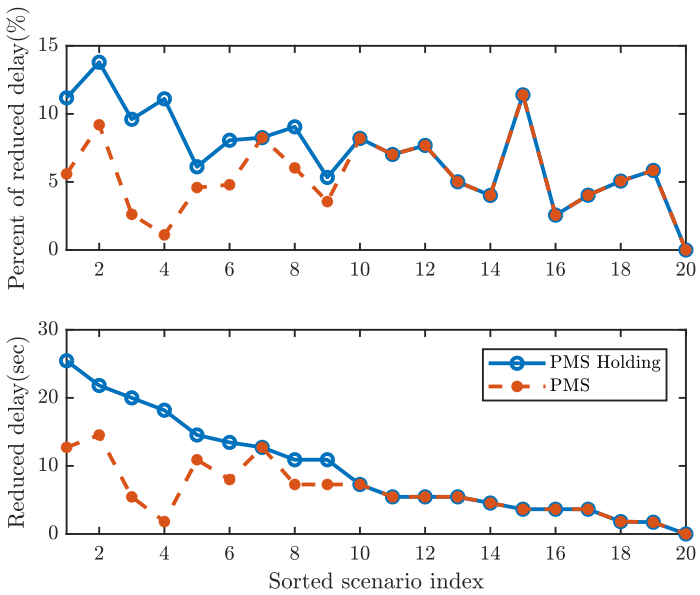


Figure 5.9: Monte Carlo simulation delay result. (PMS algorithm succeeds.)

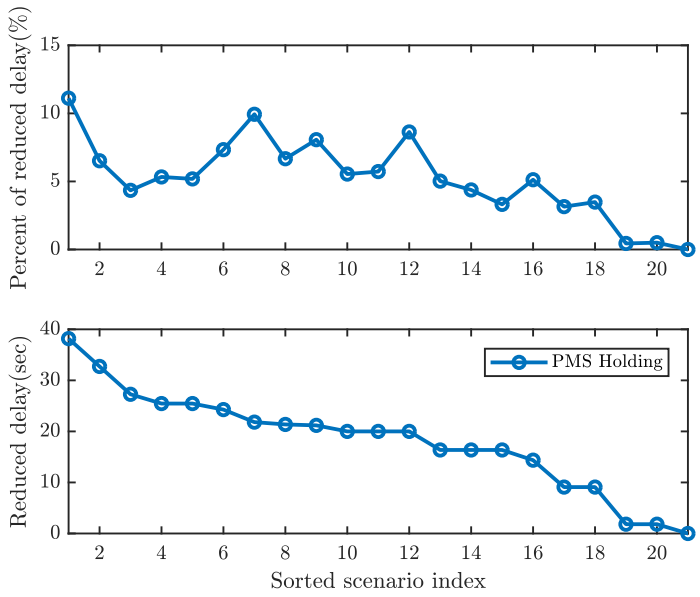


Figure 5.10: Monte Carlo simulation delay result. (PMS algorithm fails.)

Table 5.5: Monte Carlo simulation result. (PMS algorithm succeeds.)

Average result(sec)	High density			Medium density			low density		
	PMS Holding	PMS	FCFS	PMS Holding	PMS	FCFS	PMS Holding	PMS	FCFS
Average delay	61.73	62.82	64.81	53.20	53.37	53.98	53.71	53.83	54.63
Average interval	202.40	202.43	202.46	343.49	343.49	343.49	357.03	357.18	358.18
Computation time	0.2914	0.0170	0.1413	0.0573	0.0122	0.0380	0.0318	0.0096	0.0232

Table 5.6: Monte Carlo simulation result. (PMS algorithm fails.)

Average result(sec)	High density		Medium density		low density	
	PMS Holding	FCFS	PMS Holding	FCFS	PMS Holding	FCFS
Average delay	285.57	295.65	278.40	283.31	345.17	349.09
Average interval	222.34	223.14	327.11	327.70	369.43	374.38
Computation time	0.9908	0.1246	0.0640	0.0388	0.0332	0.0250

Table 5.7: Monte Carlo simulation result according to m_{max}.

Result	$m_{max} = 1$	$m_{max} = 2$	$m_{max} = 3$
Intervention rate(%)	16	1	0
Computation time(sec)	0.3049	0.3958	0.5572

5.2 Simulation Under Uncertainty

5.2.1 Simulation Settings

Numerical simulation is performed for the scheduling problem in the PMS considering uncertainties of the ETA and CDA in this section. The performance of the proposed robust scheduling algorithm is demonstrated by performing Monte Carlo simulation. For simulation, scenarios are generated based on the historical data of CJU. The aircraft category of flight is assigned 90% heavy and 10% Large. The flights through SELIN are 9.73% (r_2), and other flights enter through DANBI (r_1). Total of 30 flights exist in each scenario, and the scenario length is set to 1 hour. The scheduling is performed every 5 minutes. Overall 10 scenarios are generated and used for Monte Carlo simulation. For each scenario, 100 Monte Carlo simulations are performed.

The parameters of the scheduling algorithm are as follows. The bounds of scheduled time of arrival (STA) according to ETA, T^E and T^L, are set to -1 and 3 minutes, respectively. The maximum leg delay, $\Delta T_r^{leg,max}$, is 409 sec. for r_1 and 245 sec. for r_2 based on the length of the sequencing leg and the flight speed on it. And κ, a confidence level of robust scheduling, is 0.25. Note that the robustness is improved as κ decreases, but the feasible region of a robust solution narrows. Therefore, a suitable κ is recommended for simulation. $\tilde{T}^{CDA}$ is assumed to follow $N(275, 28^2)$. The size of the sliding window, T_W, is 20 minutes, and the freezing time, T^F, is set to 10 minutes.

Five different scheduling algorithms are used to compute the scheduling algorithms and to analyze the effect of uncertainty. The first algorithm is a normal scheduling algorithm explained in Section 3.1, which ignores the uncertainty. It is notated as '*NS*' in the following section. The second algorithm is

a robust scheduling algorithm, only considering ETA uncertainty. Thus, Eqs. (3.9)-(3.14), (4.11) are used, which is noted as '*RS 1.*' The third algorithm is a robust scheduling algorithm considering ETA uncertainty only, which is called '*RS 2.*' *RS 2* is different from *RS 1* that the time-varying characteristics of ETA uncertainty is ignored in this algorithm. The standard deviation of ETA uncertainty is fixed at the value of the uncertainty model at 15 min (39.95 sec). The fourth algorithm is a robust scheduling algorithm, only considering CDA uncertainty. Among the formulation of the robust scheduling algorithm, Eq. (4.11) is substituted for Eq. (3.15) in this algorithm, which is notated as '*RS 3.*' The last algorithm is a robust scheduling algorithm explained in Section 4.2, considering ETA uncertainty and CDA uncertainty, which is noted as '*RS 4.*' Above 5 algorithms are summarized in Table 5.8.

All simulations are conducted with a desktop PC (I7-7700 Intel Core processor and 40 GB memories). MATLAB is selected as the simulation software, and CPLEX is used to solve the MILP problem. [59, 60]

Table 5.8: Summary of scheduling algorithms.

Algorithm	Considered uncertainty	Formulation	ETA uncertainty model
NS	-	Eqs. (3.9)-(3.15)	-
RS 1	ETA	Eqs. (3.9)-(3.14), (4.11)	Table 4.1
RS 2	ETA	Eqs. (3.9)-(3.14), (4.11)	$\sigma_C = 39.95\ sec$
RS 3	CDA	Eqs. (3.9)-(3.14b), (3.15), (4.10)	-
RS 4	ETA and CDA	Eqs. (3.9)-(3.14b), (4.10)-(4.11)	Table 4.1

5.2.2 Example of Scheduling

Figures 5.11 and 5.12 show an example of the proposed scheduling algorithm. The result of the normal scheduling algorithm ignoring uncertainty (*NS*) is shown in Fig. 5.11, and the result of the robust scheduling algorithm only considering the ETA uncertainty (*RS 1*) is shown in Fig. 5.12. Because the scheduling is performed for every time step, the scheduling result at a specific time step is only provided. The schedule at the initial points (DANBI, SELIN), the turning point (p_T), and the merge point (HANUL) are represented with markers. The round marker denotes the schedule at the previous time step (t_{k-1}), and the asterisk marker denotes the schedule at the current time step (t_k). The green color denotes the flight came from PC735, the blue color represents the flight from PC731, and the magenta color denotes the flights came from MAKET. There also exist black dashed lines, which are the rescheduled STA at t_k. If the solid line and the dashed line are completely overlapped, it means that the original STA of the flight is maintained after a rescheduling. The sliding window is represented as a red region, and the flights outside the sliding window are excluded from the scheduling process at t_k.

In Fig. 5.11, the STAs of 5 aircraft are changed in rescheduling by *NS*.

(AC 3, AC 5, AC 6, AC 7, and AC 9) For AC 3, only the STA at the merge point is rescheduled because of the CDA uncertainty, while the STAs of other aircraft are totally rescheduled by the ETA uncertainty. It may imply that the ETA uncertainty has a more impact on the scheduling result than the CDA uncertainty. However, the results are slightly different for *RS 1*. The number of rescheduled flights is 3. (AC 3, AC 6, and AC 7) The STAs of AC 5 and AC 9 are maintained by the robust scheduling algorithm through the additional buffer under the uncertainties. In this example, the number of rescheduling by constraint violation is 26 for *NS* and 14 for *RS 1* during the simulation. and the average delay is 295.4 sec. for *NS* and 283.5 for *RS 1*. Thus, *RS 1* shows better performance in terms of the number of constraint violation and the average delay in this particular example. This tendency is statistically verified through Monte Carlo simulation shown in the following section.

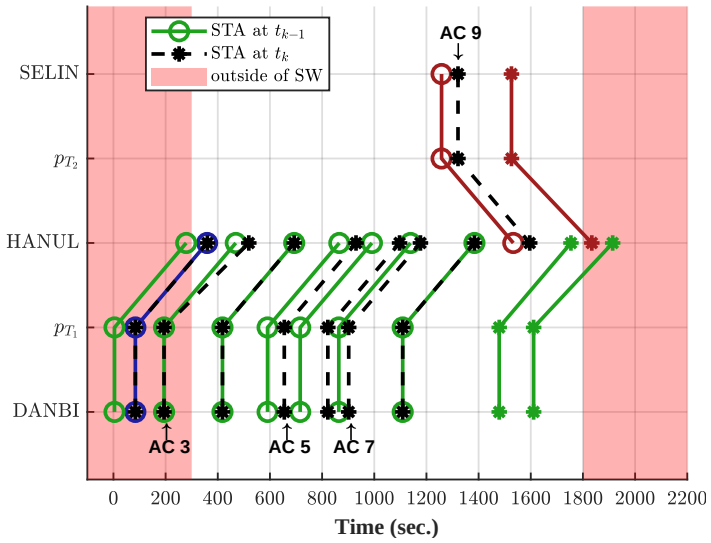


Figure 5.11: Example of normal scheduling algorithm (*NS*).

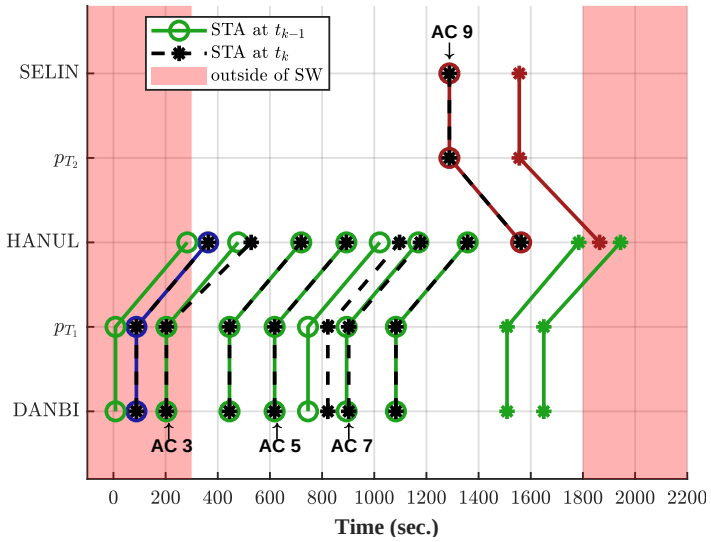


Figure 5.12: Example of robust scheduling algorithm (*RS 1*).

5.2.3 Monte Carlo Simulation

Table 5.9 shows the Monte Carlo simulation results of five scheduling algorithms. The performance of each algorithm is summarized based on the 3 main quantities: Number of constraint violations, the average delay of flights, and the amount of schedule change (ΔSTA). The number of constraint violations implies the impact of uncertainty on the flight schedules. In Table 5.9, the number of constraint violations at p_I and p_F as well as the total number of constraint violations are shown. The number of constraint violations at p_I can explain the effect of ETA uncertainty, and the number of constraint violations at p_F can explain the effect of CDA uncertainty. The total number of constraint violations is the sum of those two numbers of constraint violations, which shows the overall impact of ETA and CDA uncertainty. Figure 5.13 shows the total number of constraint violations for each algorithm.

The average delay of the flights is closely related to the performance index of the optimization problem. Equation (3.9), the sum of STA at p_F, can be converted into the total delay of a scenario by subtracting ETA from itself. Thus, it is possible to analyze the performance degradation of the robust algorithm, which sacrifices the performance index to guarantee the robustness of the solution. Figure 5.14 shows the average delay for each algorithm.

The amount of schedule change, ΔSTA, is also analyzed. Scheduling change is undesirable, because flight may fail to obey the changed schedule due to the limitation of aircraft performance, or the workload of air traffic controllers could increase much to inform each aircraft of the changed schedule. In this perspective, the small amount of schedule change is favorable. The amount of schedule change for *NS* and *RS* algorithms are shown in Fig. 5.15.

Table 5.9: Monte Carlo simulation results.[a]

	NS	*RS 1*	*RS 2*	*RS 3*	*RS 4*
# of violations at p_I	15.10	8.01	6.53	15.10	8.01
	(2.70)	(2.35)	(2.17)	(2.70)	(2.35)
# of violations at p_F	11.06	11.07	11.34	5.13	5.25
	(2.72)	(2.69)	(2.67)	(1.97)	(2.00)
Total # of violations	26.15	19.07	17.88	20.22	13.25
	(3.52)	(3.33)	(3.24)	(3.24)	(3.03)
Average delay (sec)	283.89	282.84	286.32	300.58	299.71
	(8.97)	(8.89)	(8.96)	(8.78)	(8.72)
# of nonzero ΔSTA	87.55	64.47	57.62	86.23	71.85
	(14.44)	(16.44)	(16.22)	(14.39)	(15.42)
ΔSTA per nonzero ΔSTA (sec)	61.18	50.72	48.29	62.88	46.74
	(3.77)	(4.07)	(3.99)	(3.64)	(3.69)
Total ΔSTA (sec)	5,356.53	3,269.75	2,782.52	5,422.22	3,357.87
	(931.46)	(959.31)	(928.96)	(925.83)	(947.16)

[a]The average μ and standard deviation σ are represented in 'μ (σ)'

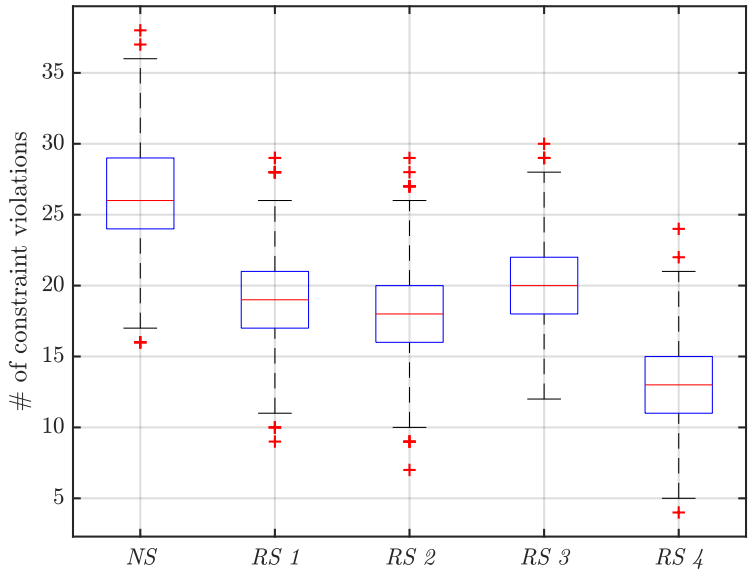


Figure 5.13: The total number of constraint violations in Monte Carlo simulation.

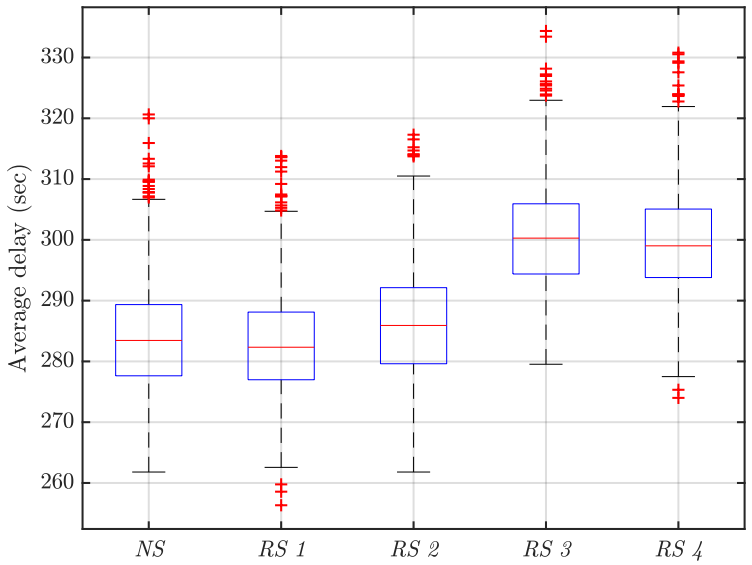


Figure 5.14: The average delay in Monte Carlo simulation.

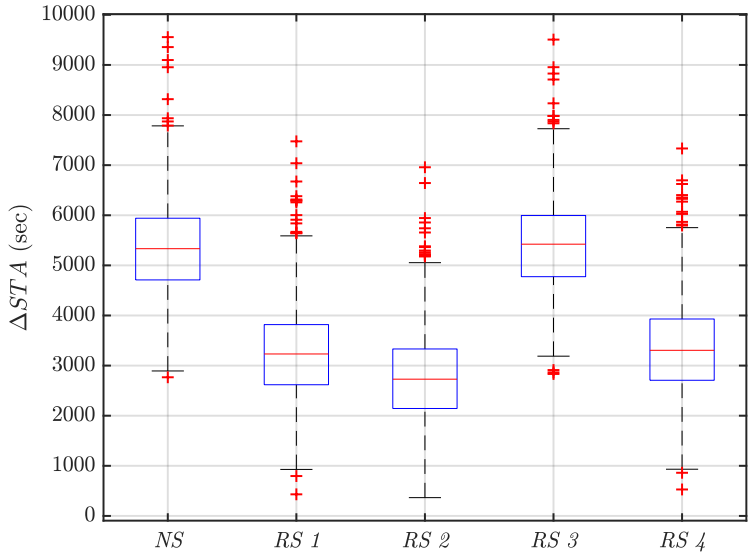


Figure 5.15: The amount of schedule change in Monte Carlo simulation.

Robust Scheduling vs. Normal Scheduling

By comparing robust scheduling algorithms (*RS 1-4*) with normal scheduling (*NS*), it is seen that the *RS* algorithms provide a meaningful decrease in the number of constraint violations. Total delay of *RS*, however, is slightly higher than that of *NS* except *RS 1*. It can be explained by the concept of robust scheduling. The robust optimization assigns an additional buffer to guarantee a robust solution, which makes the solution sacrifice the optimality to some extent. Thus, performance degradation is inevitable when using robust optimization, i.e., a more conservative solution is generated. On the other hand, it shows that the number of constraint violations also affects the average delay. As described in **Algorithm 1**, there exists a penalty for rescheduled flights. The average delay could decrease as the number of constraint violations decreases, which is related to the number of rescheduled flights. Note that a more robust solution assigns larger additional buffers to the flights, which provides the smaller number of constraint violations. The additional buffer can increase the average delay by the amount of additional buffer and decrease the average delay by the reduced number of constraint violations. Numerical simulation shows that the effect of additional buffer is superior than the number of constraint violations for *NS* and *RS 2-4*, whereas the effect of the number of constraint violations is superior than that of additional buffer for *RS 1*.

Time-varying ETA Distribution vs. Time-invariant ETA distribution

RS 1 uses time-varying ETA model, and *RS 2* uses static ETA distribution. The effect of the ETA model can be observed in Figs. 5.13-5.15. For three main indices, the performance of *RS 1* is better than that of *RS 2* in the average delay,

and worse in the number of constraint violations and the amount of schedule change. It can be analyzed that *RS 1* is less robust than *RS 2* and more efficient in average delay. The only difference between the two algorithms is the model of ETA uncertainty, and therefore it can be stated that the performance of robust scheduling depends on the modeling of uncertainty distribution. Also, the buffer assigned by the robust scheduling algorithm can be appropriately chosen by proper modeling of the uncertainty.

ETA uncertainty vs. CDA uncertainty

Let us compare the result of *RS 1*, representing ETA uncertainty, with that of *RS 3*, representing CDA uncertainty. It can be concluded that the ETA uncertainty has more influence on the flight schedule than the CDA uncertainty. Though the decreased amount of constraint violation is similar, *RS 1* shows significantly lower values than *RS 3* in average delay and the amount of schedule change. Moreover, it seems that the amount of schedule change mainly depends on the ETA uncertainty, because the algorithms *RS 1* and *RS 4* considering ETA uncertainty show similar ΔSTA, and the algorithms *NS* and *RS 3* not considering uncertainty provide similar results in terms of ΔSTA. The reason is that the ETA uncertainty affects the flight schedule for a longer time than the CDA uncertainty does. Therefore, it can be stated that the amount of schedule change is closely related to the number of constraint violations at p_I rather than the total number of constraint violations. These results support that ETA uncertainty should be considered in a scheduling problem.

RS 1 and RS 3 (single uncertainty) vs. RS 4 (double uncertainty)

RS 4 simultaneously considers CDA and ETA uncertainties, and *RS 1* and *RS 3* consider only one uncertainty, which leads to the performance difference of the algorithms. *RS 4* shows similar results in terms of ΔSTA, lower performance in average delay, and better result in the total number of constraint violations. That is, *RS 4* provides the lowest number of constraint violations but sacrifices the average delay by considering both uncertainties. It implies that considering all uncertainties may not provide the best solution for every situation. Thus, the choice between *RS 1*, *RS 3*, or *RS 4* remains as user preference. For example, if the air traffic controller prefers little constraint violation, *RS 4* is appropriate.

In summary, considering the ETA uncertainty could reduce the amount of the schedule change while maintaining the average delay. On the other hand, the robust scheduling considering the CDA uncertainty can improve the satisfaction rate of the safe separation constraint at the merge point, exploiting the average delay and the amount of schedule change. Note that the number of constraint violations and the amount of schedule change reflect the robustness of scheduling. However, they show slightly different tendencies because the amount of schedule change depends mainly on the ETA uncertainty. Considering that the communication load of air traffic controllers is related to the total number of constraint violations, the number of constraint violations is more appropriate to represent the robustness of scheduling than the amount of schedule change.

To investigate the effect of the ETA uncertainty, Monte Carlo simulation

is performed for the situation that the ETA uncertainty is decreased to 50% of the current level. Table 5.10 summarizes the simulation result for the 50% ETA uncertainty level. To compare the result of this case with that of the previous case, three main indices are shown in Figs. 5.16-5.18. For all quantities, the scheduling performance is slightly improved. In Fig. 5.16, the number of constraint violation is almost same as that of the 100% ETA uncertainty case. Comparing Table 5.9 with Table 5.10, it can be seen that the decrease of constraint violation mainly results from the constraint violation at p_I, which is closely related to the ETA uncertainty. The average delay is also decreased as much as 15 sec. on average. Note that different from the 100% ETA uncertainty case, the average delay of *RS 2* is smaller than that of *NS* for the 50% ETA uncertainty case. It seems that the decreasing effect of additional buffer becomes predominant as the ETA uncertainty level decreases. However, the amount of schedule change for 50% uncertainty level is considerably improved. Figure 5.18 shows that ΔSTA is decreased at least 600 sec. and 800 sec. on average for each scenario. It is interesting to see that the decreased amount of ΔSTA is smaller for *RS 1, RS 2* and *RS 4* representing ETA-robust algorithms. It seems that for those ETA-robust algorithms, the ΔSTA already dropped 50% in 100% ETA uncertainty case, thus the additional decreased amount of ΔSTA is smaller than that of other algorithms. In summary, the amount of schedule change is expected to decrease by ETA estimation performance for the situation that the ETA uncertainty is decreased, though it has little effect on the number of constraint violations.

The effect of κ, the major parameter of robust scheduling algorithm, Eq. (4.5), is analyzed through numerical simulations. For a specific scenario, simu-

lation is performed for various κ values. Figure 5.19 shows the size of additional buffer for ETA uncertainty and the success rate of RS2 algorithm. RS2 algorithm is chosen because the size of the additional buffer remains constant for all simulations. The additional buffer is defined as $\lambda_E \sigma_E$, which decreases as κ increases. Because κ determines the allowable probability of the violation of uncertain constraints, the robust scheduling algorithm generates a more robust solution with a smaller value of κ. However, the success rate of the robust scheduling algorithm also decreases as κ decreases. The effective ETA bound shrinks as the additional buffer increases. As a result, the size of additional buffer and the success rate of robust scheduling algorithm show completely different tendency. For the scheduling-successful cases of Fig. 5.19, the simulation results of the average delay and the number of constraint violations with respect to κ are shown in Fig. 5.20. Because a smaller κ guarantees a more robust solution, the average delay increases as κ decreases, but the number of constraint violations decreases. Therefore, the performance of the robust scheduling algorithm depends on the parameter κ, which determines the size of the additional buffer. Because the average delay and the number of constraint violations have a trade-off relation, Designers can select an appropriate value of κ considering the success rate and the performance of a robust scheduling algorithm.

The computation time could be an issue if it is too large to implement the algorithm in the real operation system. Figure 5.21 shows the computation time of *RS 4* for each time step of a scenario used in the Monte Carlo simulation. *RS 4* is chosen because it has the most constraints among algorithms and is expected to require the largest computation load. The number of flights in the sliding window is shown in Fig. 5.21. As shown in Fig. 5.21, the computation

time depends on the number of flights in the sliding window. In this example, the computation time is at most 28 sec. for each iteration, and therefore it is obviously less than 5 minutes. The average computation time is 7.24 sec., and therefore it can be stated that the proposed robust scheduling algorithm can be implemented without the issue of computation load.

Table 5.10: Monte Carlo simulation results. (50% ETA uncertainty level)[a]

	NS	*RS 1*	*RS 2*	*RS 3*	*RS 4*
# of violations at p_I	13.70	7.34	5.74	13.70	7.34
	(2.48)	(2.13)	(1.93)	(2.48)	(2.13)
# of violations at p_F	11.41	11.44	11.40	5.23	5.28
	(2.77)	(2.74)	(2.65)	(2.01)	(2.06)
Total # of violations	25.11	18.78	17.15	18.93	12.62
	(3.29)	(3.16)	(3.11)	(3.09)	(2.85)
Average delay (sec)	272.30	265.81	265.56	289.02	282.48
	(6.63)	(7.24)	(7.06)	(6.69)	(7.21)
# of nonzero ΔSTA	77.40	58.08	50.45	76.72	57.91
	(10.70)	(13.38)	(13.49)	(11.42)	(13.63)
ΔSTA per nonzero ΔSTA (sec)	53.04	44.83	42.78	54.56	46.31
	(2.82)	(2.58)	(2.67)	(2.67)	(2.45)
Total ΔSTA (sec)	4,105.34	2,603.68	2,158.47	4,185.81	2,681.60
	(561.12)	(676.78)	(657.00)	(288.25)	(681.10)

[a]The average μ and standard deviation σ are represented in 'μ (σ)'

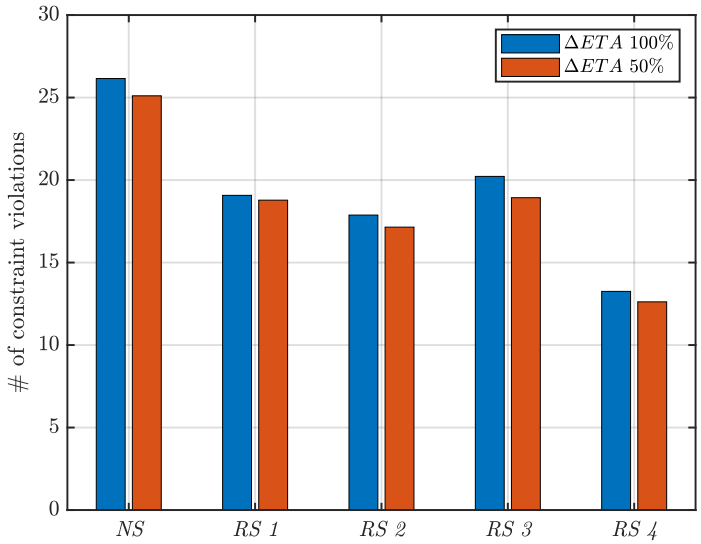


Figure 5.16: The total number of constraint violations in Monte Carlo simulation for 50% and 100% ETA uncertainty level.

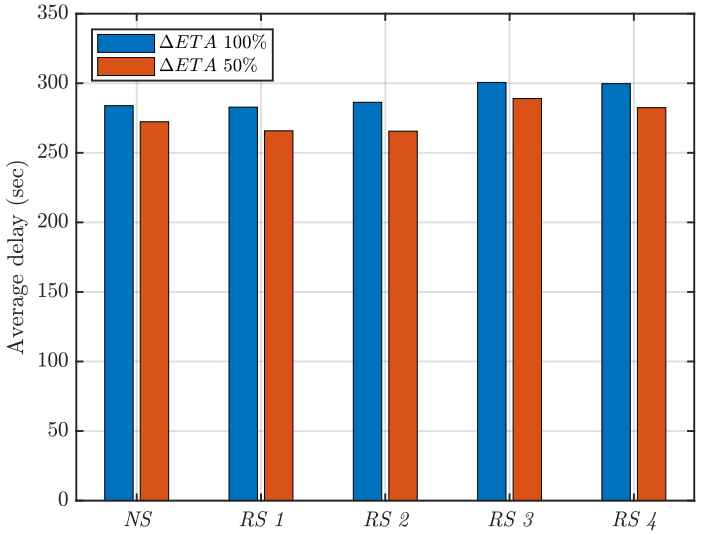


Figure 5.17: The average delay in Monte Carlo simulation for 50% and 100% ETA uncertainty level.

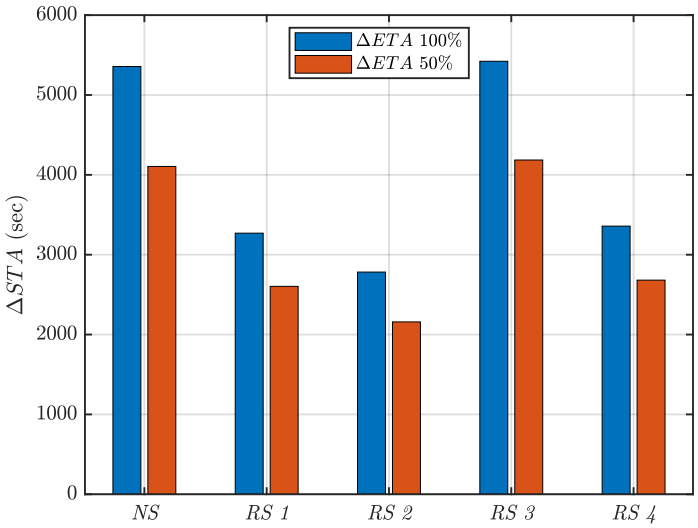


Figure 5.18: The amount of schedule change in Monte Carlo simulation for 50% and 100% ETA uncertainty level.

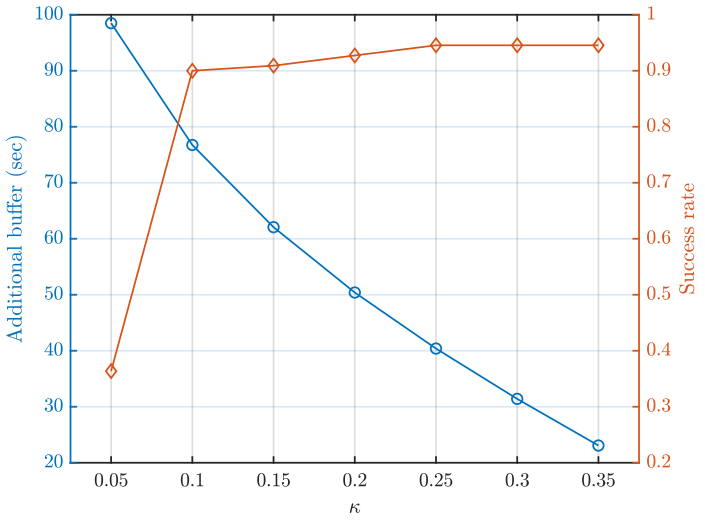


Figure 5.19: Additional buffer size and success rate w.r.t. κ.

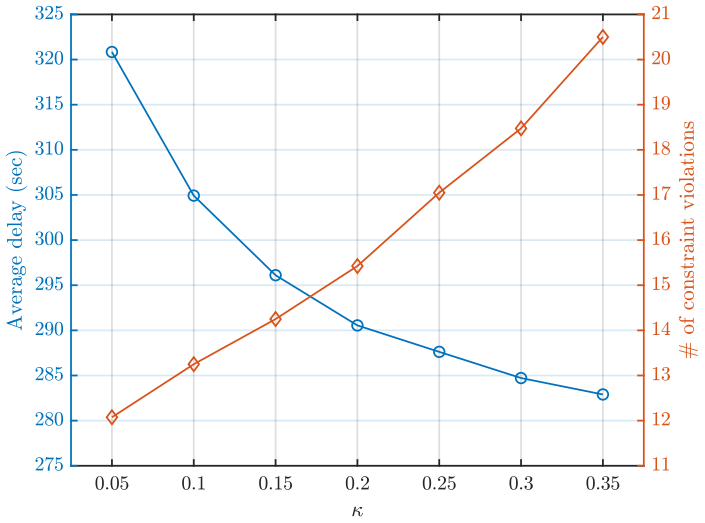


Figure 5.20: The average delay and the number of constraint violations w.r.t. κ.

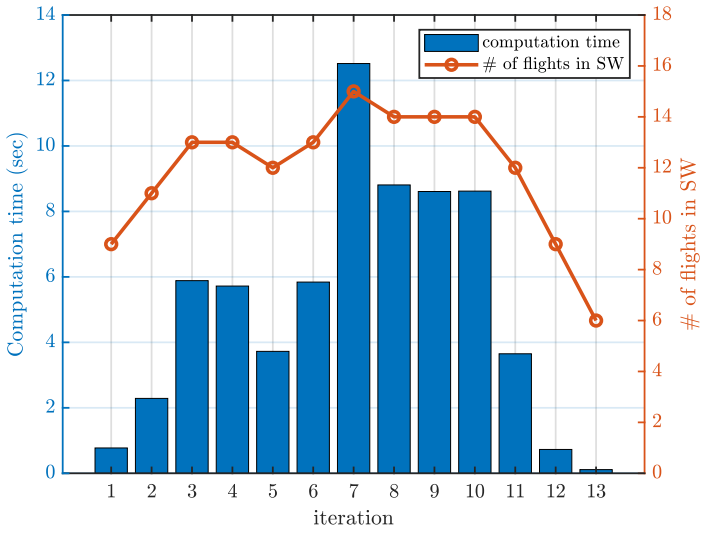


Figure 5.21: Computation time and the number of flights in sliding window of *RS 4*.

Chapter 6

Conclusions

6.1 Concluding Remarks

Scheduling algorithms in the Point Merge System (PMS) were proposed, and two main problems were addressed, specifically, a scheduling problem including a holding pattern, and a scheduling problem considering uncertainties.

The concept of the PMS and a holding pattern were explained. The airspace around the target airport, Jeju international airport (CJU), was considered. The arrival procedure and air traffic flow were presented based on the existing literature and aeronautical information publications.

First, a scheduling algorithm in the PMS without considering uncertainties was established. This PMS scheduling algorithm considering a holding pattern (PMS holding algorithm) was described based on the previous work. The PMS configuration was transformed to a node–link structure, and the optimization problem was formulated using binary and integer variables. However, the conventional MILP formulation cannot be used for the transformed node-link structure because a holding pattern involves specific characteristics that are difficult to formulate. To apply the MILP formulation to the proposed PMS holding algorithm, the route structure was changed, and virtual fixes, which are conceptual, were introduced. The existing MILP formulation was modi-

fied, and suitable constraints such as first-in-first-out and discrete holding delay constraints were introduced to incorporate the characteristics of the holding procedure into the formulation.

Second, robust scheduling algorithms of the PMS were proposed. The methodology of a robust optimization was explained. If uncertain parameters of linear programming follow normal distributions, the uncertain constraint can be converted to a deterministic constraint through the chance constraint. In addition, the ETA and CDA uncertainty models were assumed to follow a normal distribution, reference data were used to approximate the distribution. In contrast to that of the CDA uncertainty, the probability distribution of the ETA uncertainty depended on the remaining flight time, and the previous and subsequent ETA uncertainties were correlated. These characteristics of the ETA uncertainty were applied to the robust scheduling algorithm by considering a multivariate normal distribution. To perform the simulation, the sliding window technique was used to reduce the computation load.

Numerical simulations were performed to evaluate the proposed algorithms. To demonstrate the performance of the PMS holding algorithm, the algorithm was compared with the PMS and FCFS algorithms. The simulation parameters were selected by considering aeronautical information. The Monte Carlo simulation results revealed that the proposed PMS holding algorithm can likely schedule scenarios that cannot be processed using a PMS algorithm that does not consider a holding pattern. In addition, the PMS holding algorithm can realize scheduling with an additional degree of freedom owing to the consideration of the holding pattern. Consequently, the proposed algorithm outperforms the considered algorithms in terms of the delay reduction.

Moreover, the performance of the proposed robust scheduling algorithm was evaluated through a Monte Carlo simulation, in terms of the number of constraint violations, average delay, and amount of schedule change. Five different scheduling algorithms were compared to analyze the effect of each uncertainty. The results indicated that the robust scheduling algorithm corresponds to fewer constraint violations and reduces the amount of schedule change with only a slight increase in the average delays. In addition, the necessity of a robust scheduling algorithm considering the ETA uncertainty was demonstrated by comparing the effects of the ETA and CDA uncertainties.

The proposed algorithms, the PMS holding algorithm and robust scheduling algorithm, could be used as decision support tools for air traffic controllers from a practical perspective to allow a reduction in the workload of human air traffic controllers.

6.2 Further Work

Integration of ETA estimation algorithm with scheduling algorithm

The integration of the ETA estimation algorithm with the scheduling algorithm is necessary. In this dissertation, only the estimation data of the estimation algorithm were imported, and the interaction between leading and trailing flights is ignored in the estimation process. In addition, as the ETA uncertainty depends on the types of aircraft, the performance of the scheduling could be improved by the integration of scheduling and estimation algorithms.

Uncertainty of holding pattern

The uncertainty of holding pattern could be considered in the robust scheduling algorithm. There may exist holding patterns at the entry point, the last point of the sequencing leg, or the merge point in the PMS. These holding patterns are operated with the PMS and can be modeled in the scheduling problem. However, there also exists uncertainty in the holding pattern. Note that the holding constraint is an equality constraint, not an inequality constraint. Thus, it is impossible to use the chance constraint approach presented in Section 4.2. In addition, the computation load is heavier with a holding pattern as shown in Section 5.1 Therefore, a new approach to address the holding uncertainty is required and could be studied in future work.

Application on other ATM problems

The feature of the proposed algorithm is that the uncertainty is assumed to follow Gaussian distribution and addressed in robust optimization. Therefore, the proposed robust scheduling algorithm can be applied to various ATM prob-

lems considering uncertainty. For example, in a runway scheduling problem, the uncertainty of taxi-time can be handled in a similar way of CDA uncertainty. In addition, the uncertainty of arrival time in a air traffic flow management could be considered as the ETA uncertainty. Therefore, several problems for ATM considering uncertainty can adopt the approach proposed in this study.

Efficiency of scheduling algorithm

A new approach is needed to reduce the computation load of the proposed robust scheduling algorithm. In the proposed robust scheduling algorithm, every flight in sliding window is considered in the rescheduling process. Taking into account all flights for scheduling is a burden on scheduling algorithm de to a large amount of computation load, which is not desirable from the practical point of view. Reducing the size of the considered flights in the rescheduling process may alleviate this problem, and therefore a study on reducing the size of flights for the scheduling will improve the efficiency of the algorithm.

Validation in realistic simulation

To verify the performance of the proposed algorithm, realistic simulation should be conducted, which include simulation using high-fidelity model and human-in-the-loop simulation. In this study, the dynamics of flight is ignored and only the schedules of the flights are determined by the scheduling algorithm. Considering the fact that flights can conduct additional maneuver to satisfy the schedule, ignoring the flight dynamics could degrade the performance of the scheduling algorithm. In addition, in this study, the holding pattern is assumed to be freely used for every flight. Air traffic controllers, however, may not prefer to use the holding pattern in scheduling except severe situation. Therefore, more

realistic model and assumption are required to reflect the practical situations.

Bibliography

[1] *Aviation: Benefits Beyond Borders*, ATAG (Air Transport Action Group), Geneva, 2016.

[2] *Aircraft Operations. Volume I - Flight Procedures (Doc 8168)*, International Civil Aviation Organization (ICAO), 5th ed., 2006.

[3] Brooker, P., "SESAR and NextGen: Investing In New Paradigms," *Journal of Navigation*, Vol. 61, No. 2, 2008, pp. 195–208.
DOI:10.1017/S0373463307004596

[4] Favennec, B., Dymmans, T., Houlihan, D., Vergne, F., and Zeghal, K., "Point Merge Integration Enabling RNAV Application and Continuous Descent - Operational Service and Environment Definition," Technical Report 2010, EUROCONTROL, 2010.

[5] Beasley, J., Sonander, J., and Havelock, P., "Scheduling Aircraft Landings at London Heathrow Using a Population Heuristic," *Journal of the Operational Research Society*, Vol. 52, No. 5, 2001, pp. 483–493.
DOI:10.1038/sj.jors.2601129

[6] Lee, H., and Balakrishnan, H., "A Study of Tradeoffs in Scheduling Terminal-Area Operations," *Proceedings of the IEEE*, Vol. 96, No. 12,

2008, pp. 2081–2095.
DOI:10.1109/JPROC.2008.2006145

[7] Lee, H., and Balakrishnan, H., "Fuel Cost, Delay and Throughput Tradeoffs in Runway Scheduling," *American Control Conference*, IEEE, Seattle, WA, June 2008.
DOI:10.1109/ACC.2008.4586858

[8] Hu, X.-B., and Di Paolo, E., "Binary-Representation-Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 2, 2008, pp. 301–310.
DOI:10.1109/TITS.2008.922884

[9] Eun, Y., Hwang, I., and Bang, H., "Optimal Arrival Flight Sequencing and Scheduling Using Discrete Airborne Delays," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 2, 2010, pp. 359–373.
DOI:10.1109/TITS.2010.2044791

[10] Zhan, Z.-H., Zhang, J., Li, Y., Liu, O., Kwok, S. K., Ip, W. H., and Kaynak, O., "An Efficient Ant Colony System Based on Receding Horizon Control for the Aircraft Arrival Sequencing and Scheduling Problem," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 2, 2010, pp. 399–412.
DOI:10.1109/TITS.2010.2044793

[11] Harikiopoulo, D., and Neogi, N., "Polynomial-Time Feasibility Condition for Multiclass Aircraft Sequencing on a Single-Runway Airport," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 12, No. 1, 2011,

pp. 2–14.
DOI:10.1109/TITS.2010.2055856

[12] Rathinam, S., Wood, Z., Sridhar, B., and Jung, Y., "A Generalized Dynamic Programming Approach for a Departure Scheduling Problem," *AIAA Guidance, Navigation, and Control Conference*, Chicago, IL, Aug. 2009.
DOI:10.2514/6.2009-6250

[13] Kim, S. H., and Feron, E., "Impact of Gate Assignment on Departure Metering," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 2, 2014, pp. 699–709.
DOI:10.1109/TITS.2013.2285499

[14] Simaiakis, I., and Balakrishnan, H., "Probabilistic Modeling of Runway Interdeparture Times," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 2044–2048.
DOI:10.2514/1.G000155

[15] Simaiakis, I., Sandberg, M., and Balakrishnan, H., "Dynamic Control of Airport Departures: Algorithm Development and Field Evaluation," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 1, 2014, pp. 285–295.
DOI:10.1109/TITS.2013.2278484

[16] Montoya, J., Rathinam, S., and Wood, Z., "Multiobjective Departure Runway Scheduling Using Dynamic Programming," *IEEE Transactions on In-*

telligent Transportation Systems, Vol. 15, No. 1, 2014, pp. 399–413.
DOI:10.1109/TITS.2013.2283256

[17] Hu, X.-B., Chen, W.-H., and Di Paolo, E., "Multiairport Capacity Management: Genetic Algorithm With Receding Horizon," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8, No. 2, 2007, pp. 254–263.
DOI:10.1109/TITS.2006.890067

[18] Chen, H., Zhao, Y. J., and Provan, C., "Multiple-Point Integrated Scheduling of Terminal Area Traffic," *Journal of Aircraft*, Vol. 48, No. 5, 2011, pp. 1646–1657.
DOI:10.2514/1.C031332

[19] Chen, H., and Zhao, Y. J., "Sequential Dynamic Strategies for Real-Time Scheduling of Terminal Traffic," *Journal of Aircraft*, Vol. 49, No. 1, 2012, pp. 237–249.
DOI:10.2514/1.C031503

[20] Chandrasekar, S., and Hwang, I., "Algorithm for Optimal Arrival and Departure Sequencing and Runway Assignment," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 4, 2015, pp. 601–613.
DOI:10.2514/1.G000084

[21] Cox, J., and Kochenderfer, M. J., "Optimization Approaches to the Single Airport Ground-Holding Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 12, 2015, pp. 2399–2406.
DOI:10.2514/1.G001081

[22] Khadilkar, H., and Balakrishnan, H., "Integrated Control of Airport and Terminal Airspace Operations," *IEEE Transactions on Control Systems Technology*, Vol. 24, No. 1, 2016, pp. 216–225.
DOI:10.1109/TCST.2015.2424922

[23] Xue, M., and Zelinski, S., "Optimal Integration of Departures and Arrivals in Terminal Airspace," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2014, pp. 207–213.
DOI:10.2514/1.60489

[24] Choi, S., Mulfinger, D. G., Robinson, J. E., and Capozzi, B. J., "Design of an Optimal Route Structure Using Heuristics-Based Stochastic Schedulers," *Journal of Aircraft*, Vol. 52, No. 3, 2015, pp. 764–777.
DOI:10.2514/1.C032645

[25] Boursier, L., Favennec, B., Hoffman, E., Trzmiel, A., Vergne, F., and Zeghal, K., "Merging Arrival Flows Without Heading Instructions," *Proceedings of the USA/FAA Air Traffic Management R&D Seminar 2007*, Barcelona, Spain, July 2007.

[26] Favennec, B., Hoffman, E., Trzmiel, A., Vergne, F., and Zeghal, K., "The Point Merge Arrival Flow Integration Technique: Towards More Complex Environments and Advanced Continuous Descent," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, SC, Sept. 2009.
DOI:10.2514/6.2009-6921

[27] Sahin Meric, O., and Usanmaz, O., "A New Standard Instrument Arrival: The Point Merge System," *Aircraft Engineering and Aerospace Technology*, Vol. 85, No. 2, 2013, pp. 136–143.
DOI:10.1108/00022661311302742

[28] Liang, M., Delahaye, D., and Maréchal, P., "A Framework of Point Merge-based Autonomous System for Optimizing Aircraft Scheduling in Busy TMA," *5th SESAR Innovation Days*, Bologna, Italy, Dec. 2015.

[29] Liang, M., Delahaye, D., Sbihi, M., and Ma, J., "Multi-layer Point Merge System for Dynamically Controlling Arrivals on Parallel Runways," *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, IEEE, Sacramento, CA, Sept. 2016.
DOI:10.1109/DASC.2016.7778098

[30] Liang, M., Delahaye, D., and Maréchal, P., "Integrated Sequencing and Merging Aircraft to Parallel Runways with Automated Conflict Resolution and Advanced Avionics Capabilities," *Transportation Research Part C: Emerging Technologies*, Vol. 85, 2017, pp. 268–291.
DOI:10.1016/j.trc.2017.09.012

[31] Hong, Y., Choi, B., Lee, K., and Kim, Y., "Dynamic Robust Sequencing and Scheduling Under Uncertainty for the Point Merge System in Terminal Airspace," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 19, No. 9, 2018, pp. 2933–2943.
DOI:10.1109/TITS.2017.2766683

[32] de Wilde, J. M., *Implementing Point Merge System Based Arrival Management at Amsterdam Airport Schiphol*, Master's thesis, Department of Aerospace Engineering, Delft University of Technology, Netherlands, Aug. 2018.

[33] Vladimirou, H., and Zenios, S. A., *Stochastic Programming and Robust Optimization*, Springer US, Boston, MA, 1997, pp. 395–447.
DOI:10.1007/978-1-4615-6103-3

[34] Ben-Tal, A., El Ghaoui, L., and Nemirovskiĭ, A. S., *Robust Optimization*, Princeton Series in Applied Mathematics, Princeton University Press, Princeton, NJ, 2009.

[35] Murça, M. C. R., "A Robust Optimization Approach for Airport Departure Metering Under Uncertain Taxi-out Time Predictions," *Aerospace Science and Technology*, Vol. 68, Sept. 2017, pp. 269–277.
DOI:10.1016/j.ast.2017.05.020

[36] Ng, K., Lee, C., Chan, F. T., and Qin, Y., "Robust Aircraft Sequencing and Scheduling Problem with Arrival/Departure Delay Using the Min-max Regret Approach," *Transportation Research Part E: Logistics and Transportation Review*, Vol. 106, Oct. 2017, pp. 115–136.
DOI:10.1016/j.tre.2017.08.006

[37] Solveling, G., Solak, S., Clarke, J.-P., and Johnson, E., "Runway Operations Optimization in the Presence of Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 5, 2011, pp. 1373–1382.
DOI:10.2514/1.52481

[38] Xue, M., and Zelinski, S., "Integrated Arrival- and Departure-Schedule Optimization Under Uncertainty," *Journal of Aircraft*, Vol. 52, No. 5, 2015, pp. 1437–1443.
DOI:10.2514/1.C032957

[39] Taylor, C., Masek, T., and Wanke, C., "Designing Traffic Flow Management Strategies Using Multiobjective Genetic Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 10, 2015, pp. 1922–1934.
DOI:10.2514/1.G000765

[40] Taylor, C., Masek, T., Wanke, C., and Roy, S., "Designing Traffic Flow Management Strategies Under Uncertainty," *Proceedings of the USA/FAA Air Traffic Management R&D Seminar 2015*, Lisbon, Portugal, June 2015.

[41] Bosson, C. S., and Sun, D., "Optimization of Airport Surface Operations Under Uncertainty," *Journal of Air Transportation*, Vol. 24, No. 3, 2016, pp. 84–92.
DOI:10.2514/1.D0013

[42] Mori, R., "Development of a Pushback Time Assignment Algorithm Considering Uncertainty," *Journal of Air Transportation*, Vol. 25, No. 2, 2017, pp. 51–60.
DOI:10.2514/1.D0069

[43] Hong, Y., Choi, B., and Kim, Y., "Two-Stage Stochastic Programming Based on Particle Swarm Optimization for Aircraft Sequencing and Scheduling," *IEEE Transactions on Intelligent Transportation Systems*,

Vol. 20, No. 4, 2019, pp. 1365–1377.
DOI:10.1109/TITS.2018.2850000

[44] Atkin, J. A. D., Burke, E. K., Greenwood, J. S., and Reeson, D., "On-line Decision Support for Take-off Runway Scheduling with Uncertain Taxi Times at London Heathrow Airport," *Journal of Scheduling*, Vol. 11, No. 5, 2008, pp. 323–346.
DOI:10.1007/s10951-008-0065-9

[45] Lee, H., *Airport Surface Traffic Optimization and Simulation in the Presence of Uncertainties*, Ph.D. Dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, Feb. 2014.

[46] Clarke, J.-P. B., Ho, N. T., Ren, L., Brown, J. A., Elmer, K. R., Zou, K., Hunting, C., McGregor, D. L., Shivashankara, B. N., Tong, K.-O., Warren, A. W., and Wat, J. K., "Continuous Descent Approach: Design and Flight Test for Louisville International Airport," *Journal of Aircraft*, Vol. 41, No. 5, 2004, pp. 1054–1066.
DOI:10.2514/1.5572

[47] Ivanescu, D., Shaw, C., Tamvaclis, C., and Kettunen, T., "Models of Air Traffic Merging Techniques: Evaluating Performance of Point Merge," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, SC, Sept. 2009.
DOI:10.2514/6.2009-7013

[48] *Instrument Flying Handbook (FAA-H-8083-15B)*, Federal Aviation Administration, 2012.

[49] *Instrument Procedures Handbook (FAA-H-8083-16B)*, Federal Aviation Administration, 2017.

[50] OAG, *Busiest Routes 2019*, OAG Aviation Worldwide Ltd, March 2019.

[51] Lee, S., Hong, Y., and Kim, Y., "Optimal Scheduling Algorithm in Point Merge System Including Holding Pattern Based on Mixed-integer Linear Programming," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 234, No. 10, 2020, pp. 1638–1647.
DOI:10.1177/0954410019830172

[52] Capozzi, B., Atkins, S., and Choi, S., "Towards Optimal Routing and Scheduling of Metroplex Operations," *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, SC, Sept. 2009.
DOI:10.2514/6.2009-7037

[53] Bianco, L., Dell'Olmo, P., and Giordani, S., "Scheduling Models for Air Traffic Control in Terminal Areas," *Journal of Scheduling*, Vol. 9, No. 3, 2006, pp. 223–253.
DOI:10.1007/s10951-006-6779-7

[54] Bennell, J. A., Mesgarpour, M., and Potts, C. N., "Dynamic Scheduling of Aircraft Landings," *European Journal of Operational Research*, Vol. 258,

No. 1, 2017, pp. 315–327.
DOI:10.1016/j.ejor.2016.08.015

[55] Tielrooij, M., Borst, C., van Paassen, M. M., and Mulder, M., "Predicting Arrival Time Uncertainty from Actual Flight Information," *Proceedings of the USA/FAA Air Traffic Management R&D Seminar 2015*, Lisbon, Portugal, June 2015.

[56] Chai, H., and Lee, K., "En-route Arrival Time Prediction via Locally Weighted Linear Regression and Interpolation," *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, San Diego, CA, Sept. 2019. DOI:10.1109/DASC43569.2019.9081637

[57] Bishop, C. M., *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York, NY, 2006.

[58] Hu, X.-B., and Chen, W.-H., "Receding Horizon Control for Aircraft Arrival Sequencing and Scheduling," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, No. 2, 2005, pp. 189–197.
DOI:10.1109/TITS.2005.848365

[59] MATLAB, *version 9.0 (R2016a)*, The MathWorks Inc., Natick, MA, 2016.

[60] *IBM ILOG CPLEX V12. 7: User's Manual for CPLEX*, Int. Business Mach. Corp., Armonk, NY, 2016.

[61] Hong, Y., Lee, S., and Kim, Y., "Bi-Objective Optimization for Aircraft Conflict Resolution Using Epsilon-Constraint Method and TOPSIS," *18th International Conference (ICCAS 2018) on Control, Automation and Systmems*, PyeongChang, GangWon, Republic of Korea, Oct. 2018.

국문초록

본 논문에서는 포인트 머지 시스템에서의 스케줄링 알고리즘을 제시하였다. 불확실성이 없는 상황에서 홀딩패턴(Holding Pattern)을 포함한 포인트 머지 시스템(Point Merge System)의 스케줄링 문제와 불확실성이 존재하는 상황에서 포인트 머지 시스템의 스케줄링 문제를 각각 고려하였다.

불확실성이 없을 때 홀딩패턴을 포함한 포인트 머지 시스템에서의 스케줄링 알고리즘은 노드 링크 구조로 모델링하였으며, 이산 변수와 정수 변수를 이용하여 최적화 문제를 공식화하였다. 홀딩패턴의 특성 때문에 기존의 노드 링크 구조 기반의 혼합정수 선형계획법을 적용할 수 없으므로, 가상의 픽스를 도입하고 노드 링크 구조를 변경하여 혼합정수 선형계획법 기반의 최적화 문제를 제시하였다. 이 과정에서 선입선출 조건과 홀딩패턴의 이산 시간지연 조건을 추가하여 홀딩패턴의 특성을 스케줄링 알고리즘에 반영하였다.

한편, 예상도착시간과 연속강하접근의 불확실성을 고려한 포인트 머지 시스템에서의 강건 스케줄링 알고리즘을 제시하였다. 강건 최적화 방법론에 따르면 정규분포를 따르는 불확실한 매개변수가 포함된 구속조건은 확률제한 구속조건을 이용하여 불확실성이 포함되지 않은 구속조건으로 변환될 수 있다. 따라서 예상도착시간과 연속강하접근의 불확실성이 정규분포를 따른다고 가정하고 모델링을 수행하였다. 예상도착시간의 불확실성은 연속강하접근의 불확실성과는 달리 확률분포가 잔여비행시간에 의존하고 이전 시간의 불확실성과 다음 시간의 불확실성 사이의 관계가 존재하므로 다변수 정규분포를 이용하여 모델링하였다. 알고리즘의 계산량을 감소시키기 위해 슬라이딩 윈도우 기법을 사용하였다.

본 논문에서 제시된 스케줄링 알고리즘의 성능을 평가하기 위해 수치 시뮬

레이션을 수행하였다. 먼저, 몬테 카를로 시뮬레이션을 통해 홀딩패턴이 포함된 포인트 머지 시스템의 스케줄링 알고리즘의 효과를 검증하였다. 시뮬레이션 결과, 홀딩패턴이 포함될 경우 관제사의 개입률이 감소하였으며, 스케줄링의 자유도가 증가함에 따라 시간 지연량에도 이득이 있음을 확인하였다. 다음으로 불확실성을 고려하는 강건 스케줄링 알고리즘의 성능을 검증하기 위해 구속조건 위반 횟수와 평균 시간지연량, 스케줄 변동량을 기준으로 분석하였다. 몬테 카를로 시뮬레이션을 수행한 결과, 불확실성을 고려하지 않은 일반 스케줄링 알고리즘에 비해 강건 스케줄링 알고리즘은 불확실성에 의해 스케줄이 변동되는 스케줄 변동량과 구속조건 위반 횟수에서 유의미한 성능 향상이 있음을 보였다.

주요어: 항공기 스케줄링, 강건 최적화, 예상도착시간, 포인트 머지 시스템, 불확실성, 홀딩패턴

학번: 2014-21893