공학박사 학위논문

Deterministic and Stochastic Optimization for Aircraft Arrival Sequencing and Scheduling under Uncertainty

불확실성하에서 항공기 도착 시퀀싱과 스케줄링을 위한 결정론적 및 확률론적 최적화

2018 년 2 월

서울대학교 대학원

기계항공공학부

홍 유 경

Deterministic and Stochastic Optimization for Aircraft Arrival Sequencing and Scheduling under Uncertainty

불확실성하에서 항공기 도착 시퀀싱과 스케줄링을 위한 결정론적 및 확률론적 최적화

지도교수 김 유 단

이 논문을 공학박사 학위논문으로 제출함

2017 년 10 월

서울대학교 대학원

기계항공공학부

홍 유 경

홍유경의 공학박사 학위논문을 인준함

2017 년 10 월

위 원 장 _____ (인)

부위원장 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

위 원 _____ (인)

Deterministic and Stochastic Optimization for Aircraft Arrival Sequencing and Scheduling under Uncertainty

Youkyung Hong

Department of Mechanical and Aerospace Engineering

Seoul National University

APPROVED:

Changdon Kee, Chair, Ph.D.

Youdan Kim, Vice-Chair, Ph.D.

Chan Gook Park, Ph.D.

Keumjin Lee, Ph.D.

Hak-Tae Lee, Ph.D.

Deterministic and Stochastic Programming for Robust Aircraft Sequencing and Scheduling under Uncertainty

A Dissertation

by

Youkyung Hong

Presented to the Faculty of 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

Supervisor : Professor Youdan Kim

February 2018

Abstract

Deterministic and Stochastic Optimization for Aircraft Arrival Sequencing and Scheduling under Uncertainty

Youkyung Hong
Department of Mechanical and Aerospace Engineering
The Graduate School
Seoul National University

As the demand for air transportation increases, air traffic congestion is becoming a critical issue in the current air traffic control system. In particular, many researchers have recognized the need for decision support tools for human air traffic controllers in the terminal area, where incoming arrivals and outgoing departures are concentrated in a limited airspace surrounding airports. Although uncertainty comes from various sources in the terminal area, only a few existing works consider uncertainty with respect to the aircraft sequencing and scheduling problem.

In this dissertation, two different robust optimization approaches for aircraft arrival sequencing and scheduling are presented that consider the uncertainty of flight time. First, robust optimization based on deterministic programming is proposed, which has a two-level hierarchical architecture. At the higher level, an extra buffer is introduced in the aircraft safe separation constraint by adopting the typical deterministic programming. The extra buffer size is analytically derived based on a deterministic robust counterpart problem. However, robust

solutions obtained at the higher level can only be implemented in restricted situations where the magnitude of uncertainty is less than a predetermined constant value. Therefore, at the lower level, to compensate for the effects of unexpected situations under a dynamic environment, robust solutions obtained at the higher level are adjusted by using a heuristic adjustment with a sliding time window.

Second, two-stage stochastic programming based on Particle Swarm Optimization (PSO) is proposed to determine less conservative robust solutions than the robust optimization based on deterministic programming. First and second stage decision problems are defined as aircraft sequencing and scheduling, respectively. PSO is utilized for a randomized search to make the first stage decision under incomplete information about uncertain parameters. A random key representation is adopted to apply PSO to a discrete aircraft sequencing problem because PSO has a continuous nature. Next, the second stage decision is made by solving a mixed integer linear programming problem after the realization of uncertain parameters.

The performances of the two proposed robust optimization methodologies are verified through numerical simulations with historical flight data. Monte Carlo simulations are also performed for randomly generated air traffic situations.

Keywords: Aircraft sequencing and scheduling, Robust optimization, Mixed integer linear programming, Deterministic programming, Two-stage stochastic programming

Student Number: 2012-30185

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Motivation

Continuous growth of air traffic has created many problems, including severe air traffic congestion and a heavy workload for air traffic controllers. In particular, the terminal area has been considered the most complex type of airspace, where incoming arrivals and outgoing departures are concentrated in a limited airspace surrounding airports. According to the Bureau of Transportation Statistics (BTS) [1], the percentage of flights arriving and departing on time were only 77.67% and 78.34%, respectively, in 2015. In BTS, a flight is considered delayed if it arrived at or departed from the gate 15 minutes or more after the scheduled arrival or departure time. For this reason, to simplify the clearance delivery procedures in the terminal area and improve the efficiency of terminal airspace operations, Standard Instrument Departure (SID) and Standard Terminal Arrival Route (STAR) are defined with a lateral profile and with level and speed restrictions along the profile. It is recommended that arriving and departing aircraft follow SID (from the take-off phase to the en-route phase) and STAR (from the en-route phase to the initial approach phase), respectively [2]. However, it has been identified that a large number of aircraft have trouble arriving at and departing from an airport on time. For this reason, human air traffic controllers often provide tactical vectoring and manual

instructions of the heading angle and/or speed changes to achieve a safe separation between aircraft. However, these additional maneuvers, which deviate from the standard routes, would significantly degrade the safety in a complex terminal area. Therefore, the need for decision support tools for human air traffic controllers has been emphasized. Numerous works are currently underway in Single European Sky ATM Research (SESAR) and the Next Generation Air Transportation System (NextGen) to provide relevant advice to human air traffic controllers, such as Arrival Managers (AMANs) [3] and Departure Managers (DMANs) [4].

The main objective of this study is to determine robust aircraft sequencing and scheduling in the terminal area under uncertainty, especially for the Point Merge System (PMS). Note that the PMS, which was recently proposed by EUROCONTROL, is a new STAR design for merging inbound traffic [5]. Following the first implementation at the Oslo International Airport in 2011, PMS is now operational in several international airports around the world, such as Incheon (2012), Paris (2013), and Hannover (2014). Figures 1.1 and 1.2 show SID and STAR in Jeju International Airport, the Republic of Korea. In the current air traffic control system, human air traffic controllers usually provide service on a First-Come First-Served (FCFS) basis. The FCFS order has several benefits: 1) it is easy to implement, 2) it reduces the workload of human air traffic controllers, and 3) it is a fair sequencing method because the landing priority is given to the aircraft that reaches the terminal area at the earliest time [6]. However, because the minimum separation requirements depend on the International Civil Aviation Organization (ICAO) wake turbulence category [7, 8], the FCFS order might result in greater spacing between aircraft. Unfortunately, it is difficult for human air traffic controllers to calculate a new aircraft sequence deviated from the FCFS order because the number of feasi-

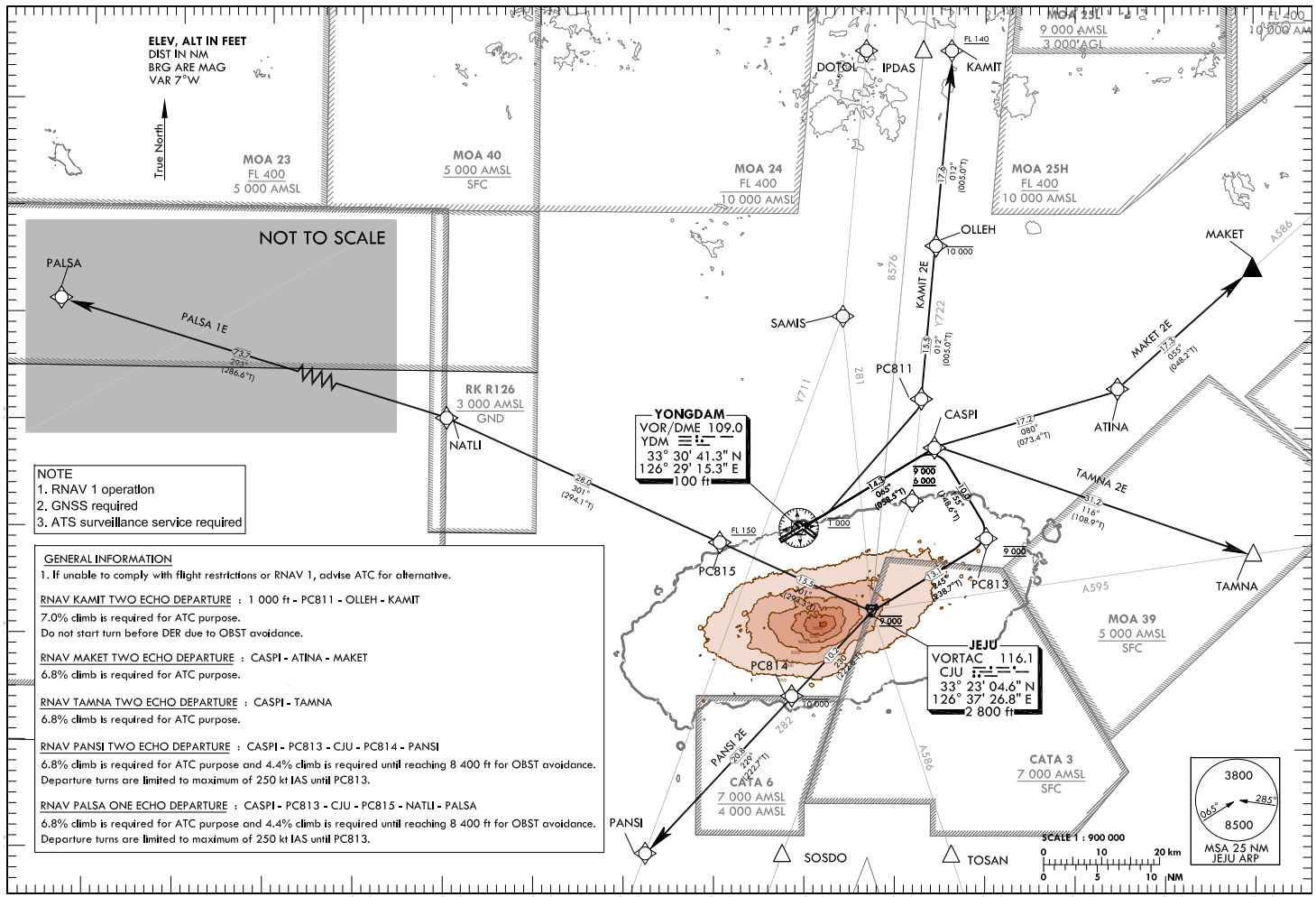


Figure 1.1 Example of SID in Jeju International Airport, the Republic of Korea

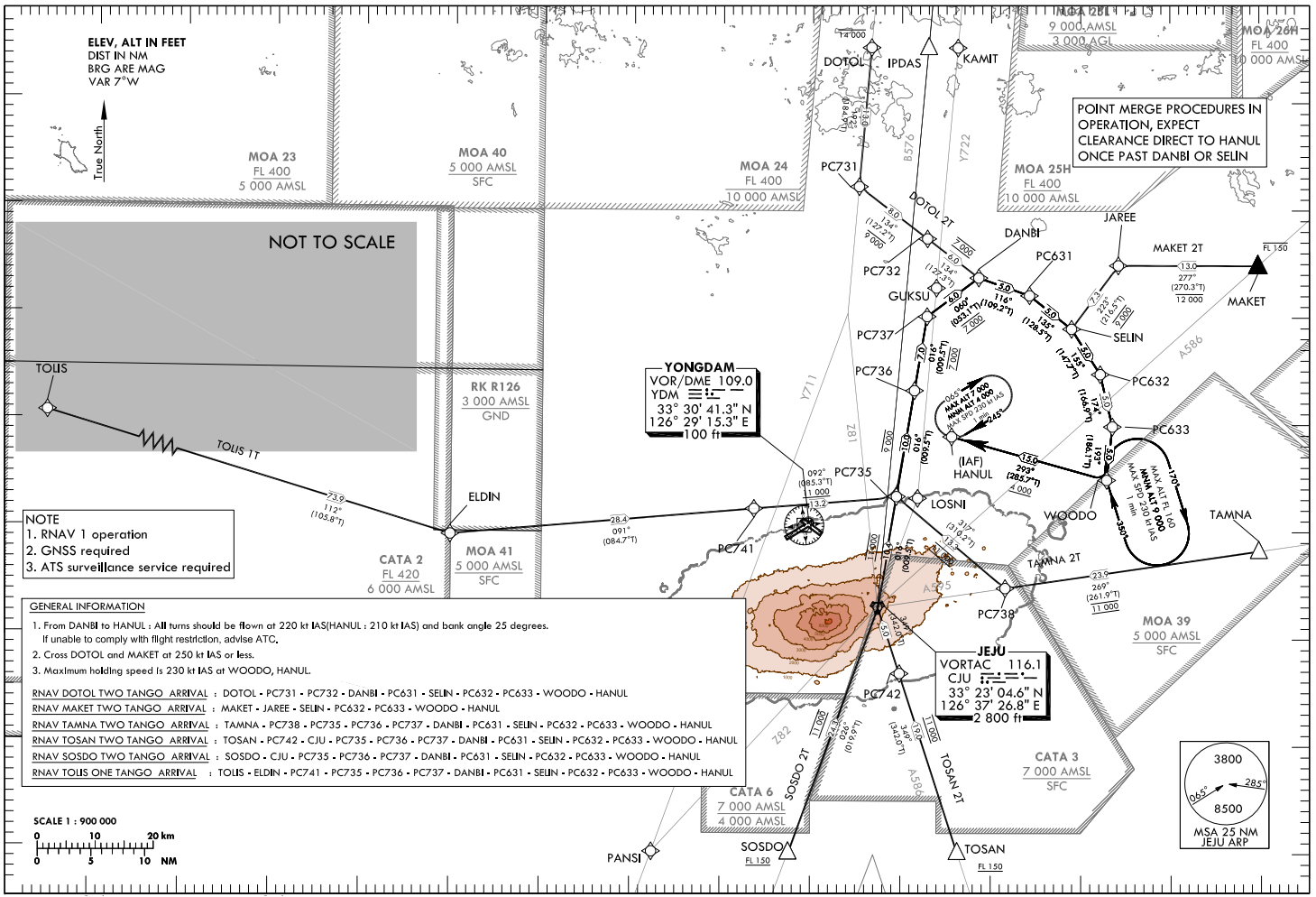


Figure 1.2 Example of STAR in Jeju International Airport, the Republic of Korea

ble sequences available for the aircraft sequencing and scheduling problem is quite large and they are faced with difficult operational considerations such as ICAO wake turbulence separation rules [9]. Therefore, when the demand for air transportation significantly increases and the traffic congestion can no longer be resolved by human air traffic controllers, it is anticipated that complementary methods, including optimization algorithms, will be required to suggest better aircraft sequences and schedules to human air traffic controllers.

It is also necessary to account for uncertainty when the aircraft sequencing and scheduling problem is solved because the uncertainty in the terminal area comes from various sources, such as weather conditions, aircraft dynamics, data availability, human factors, and interactions between arriving and departing aircraft. For this reason, the actual aircraft sequencing and scheduling is significantly different from the ideal aircraft sequencing and scheduling without considering the uncertainty. However, to date, robust optimization for the aircraft sequencing and scheduling problem to compensate for the effects of uncertainty has not received major attention in the fields of Air Traffic Management (ATM).

Studies on robust optimization can be divided into two approaches: the deterministic approach and the stochastic approach. The general strategy to address the uncertainty in the deterministic approach [10–12] is to introduce an extra buffer to the minimum separation constraint between aircraft. If the deterministic approach is focused on the aircraft conflict resolution problem, then an arbitrarily large extra buffer should be used because the highest priority is to maintain the safe separation between aircraft [12]. However, the extra buffer size should be carefully determined in the aircraft sequencing and scheduling problem because a large buffer size may increase the amount of delay, whereas a small buffer size may increase the frequency of human controller interven-

tions to resolve potential conflicts [13]. Therefore, it is required to determine the extra buffer size analytically by deriving a deterministic robust counterpart problem, which has been applied in the fields of computer and chemical engineering [14–16].

Robust optimization based on the deterministic approach may result in unnecessarily excessive flight time of each aircraft because the extra buffer added in the aircraft sequencing and scheduling problem increases the relative distance between aircraft. To determine less conservative robust solutions, uncertainty can be considered as a probability distribution function in the stochastic approach. The most common stochastic approach is a two-stage stochastic programming with discrete scenarios corresponding to realizations of uncertainty. However, it is difficult to determine robust solutions based on two-stage stochastic programming within a reasonable computation time [17] because a large number of scenarios exist, even for a small number of aircraft. Various approximation and decomposition methods have been used to solve robust optimization of large-scale systems by reducing the computational load; nevertheless, these efforts to obtain a reliable approximation and decomposition are excessive and optimality gaps are inevitable [18–21]. Therefore, in this study, some insights will be found from the two-stage stochastic programming based on evolutionary algorithms [22, 23] to compensate for the weakness of the existing algorithms.

1.2 Literature Review

The literature review presented in this chapter is divided into three main sections that cover previous research closely related to this study. Previous studies on aircraft sequencing and scheduling for the runway and terminal area are reviewed in Section 1.2.1. The literature considering uncertainty in the fields of ATM can be classified into two categories: deterministic and stochastic programming. The literature reports on deterministic and stochastic programming are reviewed in Section 1.2.2 and 1.2.3, respectively. Because previous studies on robust optimization for the aircraft sequencing and scheduling problem are few in number, some notable studies on robust optimization in the fields of computer and chemical engineering are also reviewed.

1.2.1 Aircraft Sequencing and Scheduling

A considerable amount of literature has been published on aircraft sequencing and scheduling for the runway and terminal area. The runway has been recognized as the main bottleneck of aircraft operations at airports [24]; therefore, the major aim of various runway scheduling algorithms is to maximize runway throughput (equivalent to minimizing the makespan or the landing time of the last aircraft) for arrival-only or departure-only operations [25–28].

Traditionally, the runway arrival problem has been modeled based on a job shop scheduling problem where runways and aircraft are regarded as machines and jobs, respectively [29–31]. Beasely et al. proposed a mixed-integer formulation for the single and multiple runway aircraft arrival problems, and the deviation from the estimated arrival time was minimized by using linear programming-based tree search [32]. This study was also extended to utilize Genetic Algorithm (GA), which is one of metaheuristic optimization algorithms,

to obtain sub-optimal solutions within a reasonable computation time [33, 34].

Additionally, there have been several attempts to apply the Constrained Position Shifting (CPS) framework to the runway arrival problem [25, 35–37]. Because human air traffic controllers usually provide service on a FCFS basis, as mentioned previously, excessive sequence change from the FCFS order might be infeasible in real airspace and might unnecessarily increase the workload of human air traffic controllers. Therefore, Dear et al. proposed the CPS method, in which the sequence deviation from the FCFS order is limited [35, 36]. In other words, by using the CPS method, an aircraft can be moved up to a specified maximum number of positions from the FCFS order. Balakrishnan and Chandran presented the CPS network for the single runway aircraft arrival problem and efficiently solved the problem in polynomial time (linearly in the number of aircraft) by using dynamic programming [25]. Recently, Hong et al. analyzed the effect of the CPS method on the total flight time and the difference between the FCFS order and the newly determined order [37].

In the terminal area, the aircraft arrival sequencing and scheduling problem has been considered as two separate problems because the arrival routes and departure routes are usually separated, and departing aircraft can be held on the ground when the traffic congestion is severe [38]. The aircraft arrival sequencing and scheduling problem should consider computational cost, which increases with the number of aircraft, points, and routes. Consequently, to find good solutions within a reasonable computation time, some researchers have utilized relaxation methods to apply linear programming techniques, and others have relied on metaheuristic optimization algorithms. Eun et al. solved the aircraft arrival problem by computing the lower bound of the cost based on the branch-and-bound algorithm with linear programming and applied a Lagrangian dual-decomposition for computational efficiency [38]. Moreover, Hu et

al. introduced the concept of receding horizon control, which is an N-step-ahead online optimization strategy, to the aircraft arrival problem and implemented several variations of GA to enable real-time implementation [39–42].

In addition, approaches exist that introduce integrated arrivals and departures to improve the efficiency of terminal airspace operations. Capozzi et al. described a MILP formulation that simultaneously solves the routing and scheduling problems for metroplex areas [43]. Note that, in metroplex areas, because two or more airports are located close enough in proximity, their operations can no longer be treated independently. In a follow-up study, a hybrid algorithm combining basic GA and MILP was proposed to efficiently solve the routing and sequencing problems and the scheduling problem, respectively [44]. In addition, Xue and Zelinski introduced a new formulation based on a nondominated sorting GA for optimal integration of arrivals and departures in the terminal area and examined three different separation strategies: the spatial, temporal, and hybrid separations [45].

However, there exist only a few previous works regarding the aircraft arrival sequencing and scheduling problem in PMS; moreover, most previous studies on PMS have only focused on performance validation [46–49]. Boursier et al. compared the performance of PMS with the current working method of managing arrival traffic, such as heading instructions by using data collection experiments [46]. In a follow-up study, Favnnec et al. assessed the performance of PMS in more complex environment and investigated the application of continuous descent from the further upstream airspace [47]. Ivanescu et al. provided a method for designing fast-time models for performance comparison between PMS and conventional vectoring [48]. Sahin Meric and Usanmaz applied PMS to the Istanbul International Airport and described the pre-implementation studies through real time simulation [49]. Likewise, much less attention has

been given to optimize actual flight operations conducted in PMS. Liang et al. introduced the framework of an autonomous PMS and utilized the simulated annealing algorithm, which is one of metaheuristic optimization algorithms, to solve the scheduling problem in the proposed PMS [50–53]. In addition, Hong et al. attempted to determine the exact optimal solutions for aircraft sequencing and scheduling in PMS by using MILP and achieved a reduction in the number of points and routes by considering the typical configuration and characteristics of PMS [37, 54].

1.2.2 Deterministic Programming under Uncertainty

Robust optimization based on deterministic approach has been studied in the fields of computer and chemical engineering [14–16]. Ben-Tal and Nemirovski claimed that optimal solutions of linear programming may become severely infeasible when the nominal data is slightly perturbed [14]. For this reason, they first introduced a robust optimization methodology for linear programming when uncertainty arises in the coefficients of the inequality constraints. Additionally, they suggested two methods of generating robust solutions for uncertain linear programming, depending on whether they treat the uncertainty affecting the data as bounded uncertainty or as bounded and symmetric uncertainty. Lin et al. extended the previous work [14] to MILP problems when uncertainty arises from both the coefficients and the right-hand-side parameters of the inequality constraints. They also formulated the deterministic robust counterpart problem to determine robust solutions when a probabilistic measurement is applied and feasibility tolerance and reliability level are given [15]. Janak et al. extended the previous works [14, 15] to additionally consider uncertainty that arises in the coefficients of the objective function and developed robust optimization techniques when uncertain parameters are described by

several known distributions, including a uniform distribution, a normal distribution, the difference of two normal distributions, a general discrete distribution, a binomial distribution, and a Poisson distribution [16].

However, in the fields of ATM, no detailed investigation of uncertainty has been performed to date. Most studies have only focused on sensitivity analysis, which investigates the stability of optimal solutions with respect to data perturbations. In other words, the previous studies have been interested in how much the optimal solution of the perturbed problem differs from the one of the nominal problem after the aircraft sequencing and scheduling problem is designed with perfect knowledge. Atkin et al. performed experiments where linearly distributed errors are injected into the taxi times given to the proposed aircraft scheduling algorithm for the take-off problem at the holding area and then evaluated how the performance of the proposed algorithm is changed [55]. Agogino and Rios analyzed the robustness of their optimization algorithm for large-scale air traffic flow management when the departure times are slightly different from the expected value in the proposed algorithm [56]. In this previous study, the uncertainty was modeled as a normal distribution function based on historical departure data analysis performed by Mueller and Chatterji [57], and then numerical simulations were performed by adding the uncertainty to the departure schedules. Recently, Xue et al. considered the extra buffer of 30 and 60 seconds in the separation constraints to perform sensitivity analysis [45] and investigated the impacts of uncertainty on delays and controller interventions through Monte Carlo simulations when aircraft arrival and departure times are perturbed [13].

Although the extra buffer size was not analytically determined yet by deriving the deterministic robust counterpart problem in the fields of ATM, the following studies can be considered as the few literature reports related to robust

deterministic programming [14–16]. Heidt et al. considered the uncertain earliest and latest aircraft arrival times for the robust runway scheduling problem and protected the minimum separation constraints by introducing extra buffer, the size of which is determined as the sum of mean value and two times standard deviation [10, 11]. Choi et al. extended the MILP problem for metroplex operations [43] by introducing an extra buffer to maintain the safe separation despite the arrival time error [12]. In this previous work, the extra buffer size was determined with a 90% confidence interval based on the normal distribution of the arrival time error.

1.2.3 Stochastic Programming under Uncertainty

In contrast to robust optimization based on the deterministic approach, the uncertain parameters in the stochastic approach are considered as a probability distribution function. Among several attempts with the stochastic approach, the most commonly applied stochastic techniques are stochastic models with recourse, namely, two- and multi-stage stochastic programs with discrete scenarios [22,58]. In a two-stage stochastic programming problem, some of the decisions (first stage variables) are made under incomplete information on uncertain parameters, and the remaining decisions (second stage or recourse variables) are then made after the realization of the uncertain parameters is known [59]. To efficiently and accurately solve a two-stage stochastic programming problem, several studies have been performed based on various decomposition methods: Bender's decomposition [60] and L-shaped algorithm [61]. Shapiro et al. proposed the Sample Average Approximation (SAA) method, which is a Monte Carlo simulation-based technique, to replace the stochastic program with a set of smaller problems based on a random sample of possible scenarios [62–65]. By using the SAA method, the expectation formulation of robust optimization

based on stochastic programming can be replaced by its sample average.

Recently, scenario decomposition has been reexamined to derive a Deterministic Equivalent Programming (DEP) problem from a two-stage stochastic programming problem. Although the DEP problem belongs to the class of MILP problems, it cannot be easily solved because the problem size increases with the number of uncertainty realizations [66,67]. Carøe and Schults proposed dual decomposition where a two-stage stochastic programming problem is decomposed into scenarios (i.e., realizations of uncertainty) by applying Lagrangian relaxation and adopted a branch-and-bound algorithm to solve the problem [68]. However, according to the work by Sand and Engell [69], considerable effort is required to calculate the lower bounds in dual decomposition. For this reason, Till et al. presented a new approach for the robust chemical batch scheduling problem, i.e., a hybrid algorithm to solve two-stage stochastic programming, where GA performs the search on the first stage variables instead of investing effort in the calculations of lower bounds, and the decoupled second-stage scenario problems are solved by using MILP [22, 23].

In the field of ATM, there are two previous studies influenced by two-stage stochastic programming. The first study involved determining robust solutions in the runway scheduling problem [58]. Solveling et al. investigated two-stage stochastic programming, where uncertainty arises from departure push-back delay, taxiing delay, and arrival prediction error [18, 58], and applied both Benders' decomposition [60] and the SAA method [62, 64, 65] to the problem. In the second study of Bosson et al. [19–21], they formulated the integrated terminal area and airport surface operations under uncertainty as a multi-stage stochastic programming by extending the optimization algorithm for integrated arrivals and departures proposed by Xue et al. [45]. In these previous works, solving several SAA problems with a smaller sample size rather than solving

one SAA problem with a large number of samples was suggested as a solution methodology.

1.3 Contributions

This study focuses on developing robust optimization for aircraft sequencing and scheduling in PMS under uncertainty. The main contributions of this study are described in the following sections.

1.3.1 Systematic Problem Formulation

As mentioned in the previous section, a typical aircraft sequencing and scheduling problem requires a large amount of computational time as the number of aircraft, points, and routes increases. Because the computational performance is one of the most important characteristics to use the aircraft sequencing and scheduling algorithm as a decision support tool in real operation, some works have been focused on metaheuristic approaches: simulated annealing algorithm, GA, and so on [44, 50–53]. However, although metaheuristic optimization algorithms are widely recognized as one of the most practical approaches, a thorough mathematical foundation for metaheuristic optimization algorithms has not been developed yet; therefore, the optimality of the solution cannot be guaranteed [70].

Contrary to the previous works, the key strength of this study is that MILP is utilized to find exact optimal solutions for aircraft sequencing and scheduling. By considering the typical configuration and characteristics of PMS, the number of points and routes can be significantly reduced. The only consideration with respect to the computational load of the proposed MILP formulation is the number of aircraft. Additionally, the computational load can be resolved by using the concept of sliding time window, where a large number of aircraft is divided into several small numbers of aircraft. In other words, the proposed algorithm does not determine the sequence and schedule for all aircraft at once

but only determines the sequence and schedule of some aircraft that are positioned in a specific time window. In summary, the proposed MILP formulation achieves a reduction in the number of variables and therefore can be efficiently implemented without significant computational effort.

1.3.2 Robust Optimization: Deterministic Programming

In general, robust optimization based on deterministic programming is quite simple to apply because the uncertain parameter is replaced by a constant value with the extra buffer [10–12]. In these previous studies, however, the size of the extra buffer was arbitrarily determined, regardless of its importance with respect to the efficiency of flight operations. In contrast, in this study, the buffer size is analytically derived based on the deterministic robust counterpart problem when the probability distribution function of uncertain data, feasibility tolerance, and reliability level are given [14–16]. Furthermore, the tradeoff between the total flight time and the robustness depending on the extra buffer size is identified through numerical simulations.

Robust solutions determined by the typical deterministic programming can only be implemented in restricted situations, where the magnitude of the actual uncertainty is less than the constant value that is used to replace the uncertain parameter in the typical deterministic programming. However, the actual uncertainty in real operation might be greater than the constant value. Therefore, in this study, when a substantial uncertainty during CDA is imposed on an aircraft, robust solutions determined by the typical deterministic programming are adjusted by using the proposed heuristic adjustment to compensate for unexpected situations under a dynamic environment.

1.3.3 Robust Optimization: Stochastic Programming

Although previous studies on robust optimization typically consider deterministic programming, it may result in an unnecessarily excessive flight time because of the extra buffer. For this reason, stochastic programming, especially the two-stage stochastic programming based on evolutionary algorithms, can be considered as an alternative approach to determine less conservative robust solutions. In this study, Particle Swarm Optimization (PSO) is adopted instead of GA for the first stage decision of the two-stage stochastic programming. The features of PSO can be summarized as follows. PSO can be easily implemented with less computational load than other stochastic optimization schemes [71, 72], and it has been empirically verified that the performance of PSO is not sensitive to the population size [73]. The solution quality of PSO does not rely on the initial population as long as they exist within the search space [74]. Furthermore, both PSO and GA belong to population-based metaheuristic optimization approaches, but PSO has the flexibility to control the balance between the global and local experience in the search space. This property enhances the search capabilities of PSO and avoids premature convergence to a local optimum. However, difficulties arise when PSO is applied to combinatorial problems, including the aircraft sequencing problem, because of the continuous nature of PSO. To solve this problem, a random key representation is introduced in this study to convert the continuous position values to a discrete aircraft sequence [75, 76]. Therefore, PSO can be successively utilized to search the first stage variables in the proposed robust optimization algorithm. Note that, to the best of our knowledge, our application of the two-stage stochastic programming based on an evolutionary algorithm is the first application of an evolutionary algorithm to the aircraft sequencing and scheduling problem under uncertainty.

1.4 Dissertation Organization

The organization of this dissertation is as follows. Chapter 2 explains the typical configuration and characteristics of PMS and describes the operational concept of the proposed optimal aircraft arrival sequencing and scheduling problem with two control elements. Additionally, the detailed problem formulation including decision variables, objective function, and several constraints is provided based on MILP.

In Chapter 3, robust optimization based on deterministic programming is presented to determine robust solutions for the aircraft sequencing and scheduling problem in PMS under uncertainty. Robust solutions are first determined based on the typical deterministic programming, and then they are adjusted through two types of algorithm enhancements for a dynamic environment; one is the heuristic adjustment, and the other one is the concept of sliding time window.

Chapter 4 proposes robust optimization based on stochastic programming as an alternative approach to solve the aircraft sequencing and scheduling problem under uncertainty. In the proposed two-stage stochastic programming based on PSO, the first and second stage decision problems are divided into aircraft sequencing and scheduling, respectively. The first and second stage decision problems are formulated based on PSO with random key representation and MILP, respectively.

In Chapter 5, the summary of this study and extensions for future research are provided.

Chapter 2

Mixed Integer Linear Programming for Aircraft Arrival Sequencing and Scheduling

This chapter presents the formulation of the aircraft arrival sequencing and scheduling problem in PMS without considering uncertainty, which will be considered as a nominal problem in chapters 3 and 4. In Section 2.1, the typical configuration of PMS will be presented, and the arrival procedure through PMS will be explained. In addition, by describing the characteristics of the PMS, the way how the proposed MILP formulation achieves a reduction in the number of points and routes will be explained. To clarify the operational concept of the proposed algorithm, two control elements, where the required delay can be absorbed, are explained in Section 2.2. Finally, the detailed problem formulation based on MILP including decision variables, objective function, and several constraints will be provided in Section 2.3.

2.1 Point Merge System (PMS)

2.1.1 Configuration of PMS

The typical structure of PMS consists of a single merge point and two parallel and opposite sequencing legs as shown in Fig. 2.1. The merge point is the last fix of PMS, where the arrival flows in different directions are merged with the safe separation. The sequencing legs are vertically and/or laterally separated and equidistant from the merge point.

2.1.2 Arrival Procedure through PMS

From the perspective of aircraft, the arrival procedure through PMS can be explained in chronological order as follows. When an aircraft leaves the en route airspace and then enters the terminal area, the route referred to as the sequencing leg is assigned. After the aircraft enters the pre-defined sequencing leg, it flies with a constant airspeed along the leg. When the safe separation

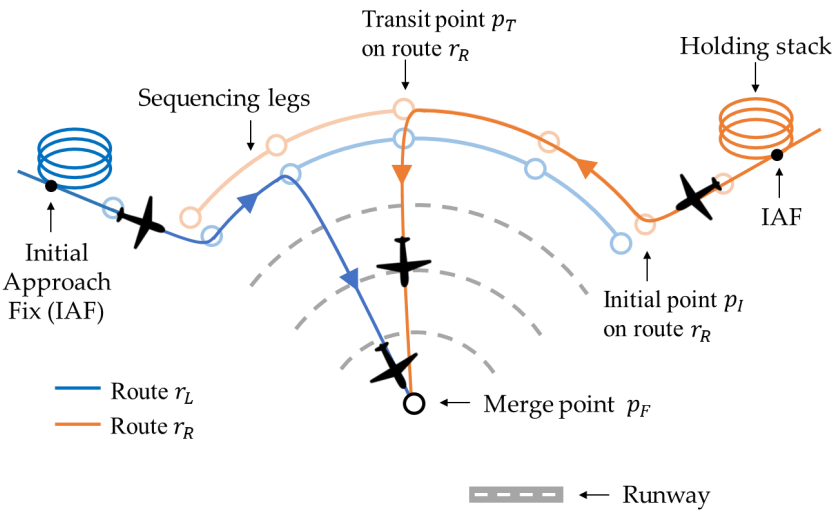


Figure 2.1 Typical structure of PMS

with the preceding aircraft in the sequence at the merge point is guaranteed, a direct-to instruction is given by human air traffic controllers. The aircraft turns off the sequencing leg and then conducts a Continuous Descent Approach (CDA) from the sequencing leg to the merge point.

2.1.3 Characteristic of PMS

The advantages and disadvantages of PMS can be summarized as follows. By providing a predefined procedure, PMS can allow human air traffic controllers to systematically manage arrival traffic. Therefore, the traffic patterns inside the terminal area can be more organized, and the safety level can be improved. Additionally, unlike a conventional approach where an aircraft descends stepwise including level flight, CDA allows an aircraft flying its individual optimal vertical profile with minimum thrust. The flight efficiency of aircraft through CDA can be improved, with reduced fuel consumption and environmental impacts [5, 77].

However, PMS might become saturated when the volume of inbound traffic through PMS is severe because the length and the amount of delay absorption of the sequencing legs in PMS are limited. When the required delay cannot be absorbed on the sequencing legs, intuitive operations should be performed by human air traffic controllers: sequencing leg run-off, aircraft vectoring and holding at the Initial Approach Fix (IAF) to absorb residual delay. However, because sequencing leg run-off and aircraft vectoring would degrade the safety inside the complex terminal area, they have been avoided in a real operation. Additionally, overreliance on the traditional circular holding stack at the IAF would increase fuel consumption, environmental impacts, and delays to passengers.

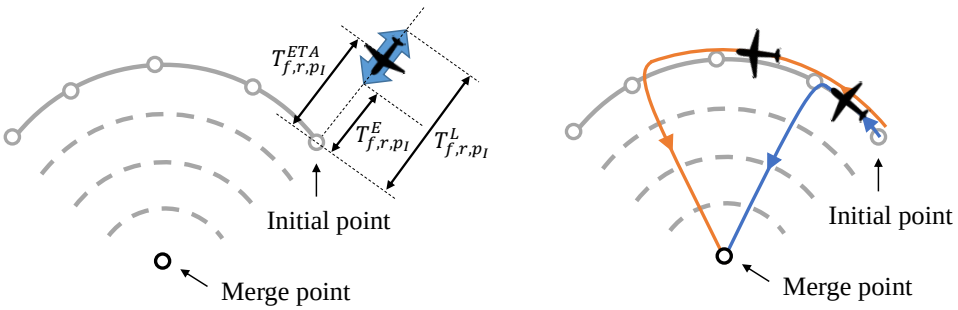
2.2 Concept of Operation

To obtain exact optimal solutions of the aircraft arrival sequencing and scheduling problem in PMS, two control elements are considered as shown in Fig. 2.2: the time of arrival into PMS and the time spent travelling along the sequencing leg. In other words, before an aircraft enters PMS, the delay can be absorbed by changing the entering time into PMS. Assuming that an aircraft follows a nominal route and speed profile [78], the estimated time of arrival at the initial point p_I, i.e., T^{ETA}_{f,r,p_I}, can be calculated by dividing the relative distance between the initial position of flight f on route r and the initial point p_I by the nominal flight speed. If an aircraft speeds up, the entering time into PMS will be earlier than T^{ETA}_{f,r,p_I}. Additionally, the entering time into PMS might be delayed by using the traditional vertical holding stack at the IAF if necessary. In this study, it is assumed that an aircraft can change its speed by less than $\pm 20\%$ with respect to the nominal flight speed. Then, the earliest and latest entering times into PMS can be determined as $0.8\, T^{ETA}_{f,r,p_I}$ and $1.2\, T^{ETA}_{f,r,p_I}$, respectively. If it is assumed that the advice is given to aircraft ten minutes before they enter PMS, then ± 2 minutes are considered for the allowed time variations for the earliest and latest arrival times, respectively.

On the other hand, after an aircraft enters PMS, the turning time on the sequencing leg can be adjusted by stretching aircraft's path on the sequencing leg to absorb the delay. In this study, it is assumed that an aircraft stays on the sequencing leg without limitation if necessary. In other words, the sequencing leg run-off procedure is allowable. Note that this assumption is operationally practical, in that the sequencing leg run-off procedure has been frequently used in a real operation because human air traffic controllers put more effort toward guaranteeing the safe separation at the merge point. The actual flight opera-

tions conducted in PMS have been determined intuitively by human air traffic controllers. However, in this study, the two control elements such as selecting optimal sequences and schedules are automatically calculated to assist human air traffic controllers in aircraft arrival sequencing and scheduling.

Note that, although 20% offset to the nominal flight speed is considered as the feasible speed range in this study, different speed ranges may be selected by considering the traffic characteristics of the target airport, which include traffic demand level, fleet mix ratio, and the capacities of arrival fixes. Actually, in previous ATM works, there is insufficient discussion on how the most feasible earliest and latest arrival times can be determined from the estimated time of arrival. According to [79], because of the resultant fuel expenditure, the time variation for the earliest time of arrival at runway is usually limited to one minute. In [50], three minutes allowed time advance and the maximum delay of ten minutes are considered for the feasible constraint of the earliest and latest arrival times, respectively.



(a) Time of arrival into PMS (b) Time spent travelling along the leg

Figure 2.2 Two control elements

2.3 Problem Formulation

2.3.1 Decision Variables

Two decision variables are defined to formulate the aircraft arrival sequencing and scheduling problem in PMS based on MILP: a continuous variable $T_{f,r,p}$ and a binary variable $S_{f,f',r,r',p}$ [43]. First, $T_{f,r,p}$ is required to solve the aircraft scheduling problem, which represents the time that a flight f reaches a point p on route r. Second, $S_{f,f',r,r',p}$ is for the aircraft sequencing problem, which becomes one when a flight f on route r is prior to a flight f' on route r' at a point p.

Note that each flight f, route r, and point p belong to the set of flights F, the set of points P, and the set of routes R, respectively. Based on the typical configuration of PMS as shown in Fig. 2.1, the sets P and R are considered as $P = \{p_I, p_T, p_F\}$ and $R = \{r_L, r_R\}$, respectively, where p_I is an entry point into the sequencing legs, p_T is a transit point on the sequencing legs, and p_F is a final merge point. Because arrival traffic flows through PMS are generally in the opposite direction as mentioned previously, the left and right traffic flows are considered, i.e., the routes r_L and r_R.

2.3.2 Objective Function

The objective function is defined to minimize the total flight time that is required for all flights to reach the merge point as follows:

$$\min \ J = \sum_{f=1}^{N_F} \sum_{r=1}^{N_R} A_{f,r} T_{f,r,p_F} \tag{2.1}$$

where N_F and N_R are the number of flights in F and routes in R, respectively, and $A_{f,r}$ is a binary parameter that takes the value one when a flight f is assigned to route r and zero otherwise.

2.3.3 Constraints

Entering Time Range Constraint

The first constraint is to define the entering time range into PMS for each flight. Based on the speed range considering 20% offset to the nominal flight speed, the time of flight f on route r at the initial point p_I, i.e., T_{f,r,p_I}, can be limited as follows:

$$A_{f,r}\left(T_{f,r,p_I} - 0.8\, T^{ETA}_{f,r,p_I}\right) \geq 0, \quad \forall f \in F, \forall r \in R \tag{2.2}$$

$$A_{f,r}\left(T_{f,r,p_I} - 1.2\, T^{ETA}_{f,r,p_I}\right) \leq 0, \quad \forall f \in F, \forall r \in R \tag{2.3}$$

Ordering Constraint

The second constraint is to determine the sequence between two flights f and f' at a shared point. Because different sequencing legs are vertically and/or laterally separated, the sequence at the initial point p_I is only determined among flights on same route. If the safe separation between any two flights can be achieved at the initial point p_I, then it can be maintained on the legs, i.e., at the transit point p_T, because they fly with a constant speed. On the other hand, at the merge point p_F, the sequence among flights should be determined whatever their assigned routes are. By considering these characteristics of PMS, the aircraft sequence at points p_I and p_F can be determined under the following constraints.

$$S_{f,f',r,r',p_I} + S_{f',f,r',r,p_I} = A_{f,r}A_{f',r'}, \quad \forall f, f' \in F, \forall r, r' \in R, f \neq f', r = r' \tag{2.4}$$

$$S_{f,f',r,r',p_F} + S_{f',f,r',r,p_F} = A_{f,r}A_{f',r'}, \quad \forall f, f' \in F, \forall r, r' \in R, f \neq f' \tag{2.5}$$

Safe Separation Constraint

The third constraint is to make the safe separation between two flights f and f' at a shared point. The constraint can be expressed as follows:

$$SEP_{f,f',p_I} \leq A_{f',r'}T_{f',r',p_I} - A_{f,r}T_{f,r,p_I} + M(1 - S_{f,f',r,r',p_I}), \quad (2.6)$$
$$\forall f, f' \in F, \forall r, r' \in R, f \neq f', r = r'$$
$$SEP_{f,f',p_F} \leq A_{f',r'}T_{f',r',p_F} - A_{f,r}T_{f,r,p_F} + M(1 - S_{f,f',r,r',p_F}), \quad (2.7)$$
$$\forall f, f' \in F, \forall r, r' \in R, f \neq f'$$

where M is an arbitrarily large number, and SEP_{f,f',p_I} and SEP_{f,f',p_F} are the minimum safe separation required at the initial and merge points, respectively. The parameters SEP_{f,f',p_I} and SEP_{f,f',p_F} can be determined according to the ICAO wake turbulence category [7] where three types of aircraft are considered: heavy, large, and small aircraft. The minimum safety separation standards defined in terms of distance are summarized in Table 2.1. By considering the flight speed, the separation distance given in Table 2.1 can be converted into the time separation [8]. In this study, it is assumed that an aircraft flies with a constant speed of 210 knots along the sequencing legs and reaches the merge point with the standard landing speed of 160 knots regardless of aircraft type.

Table 2.1 Minimum safety separation standards defined in terms of the distance [7, 8]

		Class of trailing aircraft		
		Heavy	Large	Small
Class of leading aircraft	Heavy	4 NM	5 NM	6 NM
	Large	3 NM	3 NM	4 NM
	Small	3 NM	3 NM	3 NM

For example, the minimum separation distance 3 NM can be converted into the time separations $3,600 \times (3/210) = 51.43$ seconds at the initial point and $3,600 \times (3/160) = 67.50$ seconds at the merge point.

Flight Time Constraint

The fourth constraint is to calculate the flight time of each flight between two points. The constraint can be represented as follows:

$$A_{f,r}\left(T_{f,r,p_T} - T_{f,r,p_I}\right) \geq 0, \qquad \forall f \in F,\, \forall r \in R \tag{2.8}$$

$$A_{f,r}\left(T_{f,r,p_F} - T_{f,r,p_T} - T^{CDA}\right) = 0, \qquad \forall f \in F,\, \forall r \in R \tag{2.9}$$

where T^{CDA} denotes the flight time during CDA. Note that T^{CDA} can be set to a constant value when it is assumed that an aircraft performs CDA precisely without uncertainty. Equation (2.8) presents that the flight time along the sequencing leg is greater than zero; in other words, the lower bound of flight time constraint is set to zero. However, as mentioned previously, the upper bound is not considered by assuming that an aircraft stays on the sequencing leg without limitation if necessary.

2.3.4 Mathematical Formulation

The final MILP formulation to solve the aircraft arrival sequencing and scheduling problem in PMS can be expressed as follows:

$$\min \; J = \sum_{f=1}^{N_F} \sum_{r=1}^{N_R} A_{f,r} T_{f,r,p_F} \tag{2.10}$$

subject to Eqs. (2.2)–(2.9).

Chapter 3

Deterministic Programming for Aircraft Arrival Sequencing and Scheduling under Uncertainty

In this chapter, robust optimization based on deterministic programming is presented to consider the uncertainty of flight time during CDA. The proposed deterministic programming has a two-level hierarchical architecture as illustrated in Section 3.1. Section 3.2 formulates the problem of the deterministic programming for aircraft arrival sequencing and scheduling under uncertainty. Based on the typical deterministic programming, the uncertainty of flight time during CDA is replaced by a constant value with the extra buffer, and the extra buffer size is analytically derived based on the deterministic robust counterpart problem. In Section 3.3, algorithm enhancements to be implemented in dynamic environments, i.e., heuristic adjustment and sliding time window, are explained. The deterministic programming is summarized in Section 3.4. The limitations of the traffic handling capacity of the current PMS and the probabilistic distribution function of the uncertainty of flight time during CDA are identified by performing historical flight data analysis of Jeju International Airport, the Republic of Korea, in Section 3.5. Section 3.6 presents a toy problem to clearly provide intuition about the proposed deterministic programming with a small number of aircraft. In Section 3.7, the proposed deterministic programming is validated through numerical simulation.

3.1 Hierarchical Architecture

To determine dynamic robust solutions for aircraft arrival sequencing and scheduling in PMS based on deterministic programming, a two-level hierarchical architecture is considered as shown in Fig. 3.1. In the higher level, before aircraft enter PMS, sequence and schedule are determined based on the typical deterministic programming where the extra buffer size is added to compensate for the effects of uncertainty. The outputs of the higher level are the continuous decision variable $T_{f,r,p}$ and the binary decision variable $S_{f,f',r,r',p}$.

As aircraft progress through PMS, the static robust solutions determined by the typical deterministic programming might become obsolete and be required to be updated under dynamic environments. In the lower level, if a substantial uncertainty during CDA is imposed on an aircraft f^*, then some aircraft modify their schedules determined in the higher level to compensate for the delay of aircraft f^* and to achieve the safe separation between aircraft at the merge point. Finally, the outputs of the lower level are the modified $T_{f,r,p}$, i.e., $T'_{f,r,p}$.

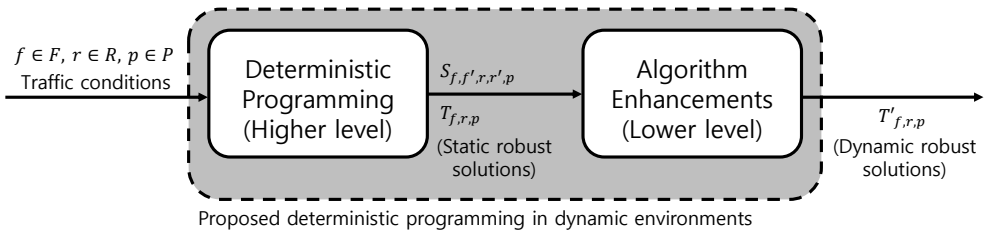


Figure 3.1 Two-level of hierarchical architecture for the proposed deterministic programming in dynamic environments

3.2 Deterministic Programming

3.2.1 Impact of Uncertainty

Let us consider the flight time constraint in Eqs. (2.8) and (2.9). In Eq. (2.9), the flight time during CDA, i.e., T^{CDA}, is set to a constant value because it is assumed that an aircraft performs CDA accurately. However, in a real operation, the CDA trajectory might deviate from the predefined path because of several reasons. A typical source of deviation in the CDA trajectory is wind disturbances, which cannot be precisely predicted. The CDA trajectory of aircraft is computed before the execution of the flight management system, which considers wind speed and direction. However, aircraft could fail to accurately follow the predefined CDA trajectory if the considered wind components differ from the actual ones. Another reason for the uncertainty of flight time during CDA is that when aircraft receive the direct-to instruction from human air traffic controller, the amount of heading changes required to leave the sequencing leg and to start the descent maneuver is different depending on the location of the aircraft along the leg.

By considering the uncertainty which arises in the CDA trajectory, Eq. (2.9) can be re-established as follows:

$$A_{f,r}\left(T_{f,r,p_F} - T_{f,r,p_T} - \tilde{T}^{CDA}\right) \geq 0, \qquad \forall f \in F, \forall r \in R \tag{3.1}$$

where $\tilde{T}^{CDA}$ represents the uncertainty of flight time during CDA.

3.2.2 Determination of Extra Buffer Size [15]

Let us consider the general MILP problem as follows:

$$\begin{aligned} &\min && c^T x \\ &\text{subject to} && Ax \leq b \end{aligned} \tag{3.2}$$

where A is a matrix with elements a_{ij}, and x is a vector with elements x_i. Assume that uncertainty exists in the right-hand-side parameter of the inequality constraint in Eq. (3.2), i.e., b_l for the l-th constraint. Then, the uncertain inequality can be represented as follows:

$$\sum_m a_{lm} x_m \leq \tilde{b}_l \tag{3.3}$$

where the uncertainty $\tilde{b}_l$ can be represented as $\tilde{b}_l = (1 + \varepsilon\, \xi_l)\, b_l$, and ε and ξ_l are a given uncertainty level and an independent random variable, respectively.

Suppose that the random variable $\xi = -\xi_l\, b_l$ has a normal distribution with zero mean and standard deviation $\sqrt{b_l^2}$. With these conditions, a vector x can be considered to be a robust solution when it satisfies the following conditions. i) x is feasible for the nominal problem, and ii) for the l-th inequality, the probability of violation of the uncertain inequality, i.e., Eq. (3.3), is at most κ, which is

$$\Pr\left\{ \sum_m a_{lm} x_m > \tilde{b}_l + \delta \max[1, |b_l|] \right\} \leq \kappa \tag{3.4}$$

where $\delta > 0$ is a given feasibility tolerance, and $\kappa > 0$ is a given reliability level. To find a robust solution x considering the above two conditions, **Theorem 1** can be derived based on the previous work [16].

Theorem 1. Given an infeasibility tolerance δ, a reliability level κ, and an uncertainty level ε, to generate robust solutions, the modified MILP problem can be defined as follows:

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & Ax \leq b, \\ & \sum_m a_{lm} x_m + \varepsilon\, \lambda \sqrt{b_l^2} \leq b_l + \delta \max[1, |b_l|], \quad \forall l \end{aligned} \tag{3.5}$$

where $\lambda = F_n^{-1}(1-\kappa)$, and F_n^{-1} is the inverse distribution function of a random variable with normal distribution, i.e., $F_n(\lambda) = \Pr\{\xi \leq \lambda\}$.

Proof. Let us consider that x satisfies the additional constraint, i.e., Eq. (3.5). Then, we have

$$
\begin{aligned}
& \Pr\Big\{ \sum_m a_{lm}x_m > \tilde{b}_l + \delta \max[1, |b_l|] \Big\} \\
= \; & \Pr\Big\{ \sum_m a_{lm}x_m > b_l + \varepsilon\, \xi_l\, |b_l| + \delta \max[1, |b_l|] \Big\} \\
= \; & \Pr\Big\{ -\varepsilon\, \xi_l\, |b_l| > b_l + \delta \max[1, |b_l|] - \sum_m a_{lm}x_m \Big\} \\
\leq \; & \Pr\Big\{ -\varepsilon\, \xi_l\, |b_l| > \varepsilon\, \lambda \sqrt{b_l^2} \Big\} \qquad (3.6) \\
= \; & \Pr\Big\{ -\xi_l\, |b_l| / \sqrt{b_l^2} > \lambda \Big\} \\
= \; & 1 - \Pr\Big\{ -\xi_l\, |b_l| / \sqrt{b_l^2} \leq \lambda \Big\} \\
= \; & 1 - F_n(\lambda) \\
= \; & 1 - (1-\kappa) \\
= \; & \kappa
\end{aligned}
$$

where $-\xi_l\, |b_l| / \sqrt{b_l^2}$ is a random variable with normal distribution. □

The deterministic robust counterpart problem in Eq. (3.5) indicates that if the optimization problem with the extra buffer $\varepsilon\, \lambda \sqrt{b_l^2}$ is solved, then the solution x is feasible at most κ under the uncertainty $\tilde{b}_l$.

Let us apply **Theorem 1** to the uncertain inequality Eq. (3.1). Suppose that the uncertainty of flight time during CDA is known to have a normal distribution with mean μ^{CDA} and standard deviation σ^{CDA}, i.e., $\tilde{T}^{CDA} \sim N(\mu^{CDA}, {\sigma^{CDA}}^2)$. Then, Eq. (3.1) can be replaced by the following inequality to obtain robust solutions to compensate for the uncertainty of flight time

during CDA.

$$A_{f,r}\Big\{T_{f,r,p_F} - T_{f,r,p_T} - \left(\mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}\right)\Big\} \;\geq\; 0, \quad \forall f \in F, \forall r \in R \quad (3.7)$$

where $\varepsilon\,\lambda\,\sigma^{CDA}$ is the extra buffer when the infeasibility tolerance δ is not allowed. For example, if a 5% reliability level κ is considered and the uncertainty level ε is set to unity, then the extra buffer is set as $1.645 \times \sigma^{CDA}$. In other words, the probability of violation of Eq. (3.7) because of the uncertainty of flight time during CDA is at most 5% with the extra buffer of $1.645 \times \sigma^{CDA}$.

Note that, although the uncertain parameter with a normal distribution is considered in this study, other probability distributions such as a uniform distribution, the difference of two normal distributions, a general discrete distribution, a binomial distribution, and a Poisson distribution can be easily extended to robust optimization based on deterministic programming [16].

3.2.3 Mathematical Formulation

Finally, the typical deterministic programming to obtain robust solutions of the aircraft arrival sequencing and scheduling problem in PMS considering the uncertainty of flight time during CDA can be expressed as follows:

$$\min \;\; J = \sum_{f=1}^{N_F}\sum_{r=1}^{N_R}\Big\{A_{f,r}T_{f,r,p_F} + A_{f,r}\left(T_{f,r,p_F} - T_{f,r,p_T}\right)\Big\} \quad (3.8)$$

subject to Eqs. (2.2)–(2.8) and Eq. (3.7). The first term of Eq. (3.8) is to minimize the total flight time required for all flights to reach the merge point. Additionally, the second term of Eq. (3.8) is to reduce the amount of delay absorption during CDA. In this study, it is assumed that an aircraft can make a detour after turning off the sequencing leg if it is necessary. Therefore, if a substantial uncertainty during CDA is imposed on an aircraft, the safe separation between aircraft at the merge point might not be guaranteed without this

assumption. However, when considering the objective of PMS and the advantage of CDA, the amount of delay absorption during CDA should be as small as possible. For this reason, the second term is incorporated into the objective function as a soft constraint.

3.3 Algorithm Enhancements for Dynamic Environments

3.3.1 Heuristic Adjustment

Throughout this dissertation, let us refer to robust solutions determined in Section 3.2 as static robust solutions. The static robust solutions are valid only if the actual flight time during CDA is less than the constant value representing the uncertainty of flight time during CDA in the typical deterministic programming, i.e., $\mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}$ in Eq. (3.7). For this reason, the static robust solutions should be adjusted to maintain the safe separation between aircraft at the merge point even if a substantial uncertainty during CDA is imposed on an aircraft. Note that the substantial uncertainty means that the actual flight time during CDA is greater than the constant value $\mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}$ in Eq. (3.7).

If a substantial uncertainty is imposed on an aircraft f^* during CDA, then its new arrival time T'_{f^*,r,p_F} at the merge point can be calculated as follows:

$$T'_{f^*,r,p_F} = T_{f^*,r,p_F} - (\mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}) \;+\; \tilde{T}^{CDA}_{f^*,r} \tag{3.9}$$

where T_{f^*,r,p_F} is the arrival time at the merge point of aircraft f^* determined in Section 3.2, and $\tilde{T}^{CDA}_{f^*,r}$ represents the actual flight time during CDA of aircraft f^*.

Now, let us adjust T_{f,r,p_I}, T_{f,r,p_T}, and T_{f,r,p_F} of other aircraft that follow the aircraft f^*. First, consider a case that an aircraft f has not yet passed the initial point of PMS when an aircraft f^* reaches the merge point. Then, T_{f,r,p_I} of aircraft f can be adjusted. Otherwise, if aircraft f has already entered PMS before aircraft f^* reaches the merge point, T_{f,r,p_I} of aircraft f continues to have the value determined by the typical deterministic programming in Eq. (3.8). It

can be summarized as follows.

$$0 \leq T'_{f,r,p_I} \leq M \qquad \text{If } T_{f,r,p_I} > T'_{f^*,r,p_F} \tag{3.10}$$

$$T'_{f,r,p_I} = T_{f,r,p_I} \qquad \text{Otherwise} \tag{3.11}$$

Second, consider a case that an aircraft f has not yet passed the transit point p_T when aircraft f^* reaches the merge point. The flight time along the sequencing legs, i.e., T_{f,r,p_T}, of aircraft f can be newly determined as shown in Eq. (3.12). Otherwise, T_{f,r,p_T} of aircraft f continues to have the value determined by the typical deterministic programming in Eq. (3.8).

$$0 \leq T'_{f,r,p_T} \leq M \qquad \text{If } T_{f,r,p_T} > T'_{f^*,r,p_F} \tag{3.12}$$

$$T'_{f,r,p_T} = T_{f,r,p_T} \qquad \text{Otherwise} \tag{3.13}$$

Lastly, if an aircraft f has not yet passed the merge point when aircraft f^* reaches the merge point, then the arrival time at the merge point of aircraft f can be determined by Eq. (3.14). Otherwise, T_{f,r,p_F} continues to have the value determined by the typical deterministic programming in Eq. (3.8), as described in Eq. (3.15).

$$0 \leq T'_{f,r,p_F} \leq M \qquad \text{If } T_{f,r,p_F} > T'_{f^*,r,p_F} \tag{3.14}$$

$$T'_{f,r,p_F} = T_{f,r,p_F} \qquad \text{Otherwise} \tag{3.15}$$

The proposed heuristic adjustment for the substantial uncertainty is finally achieved by solving Eq. (3.8) with Eqs. (3.9)–(3.15) serving as additional constraints, which can be expressed as follows:

$$\min \; J = \sum_{f=1}^{N_F} \sum_{r=1}^{N_R} \left\{ A_{f,r} T_{f,r,p_F} + A_{f,r} \left(T_{f,r,p_F} - T_{f,r,p_T} \right) \right\} \tag{3.16}$$

subject to Eqs. (2.2)–(2.8), Eq. (3.7) and Eqs. (3.9)–(3.15).

Although, in this study, it is focused on a simplified case where the substantial uncertainty is imposed on one aircraft, the proposed heuristic adjustment can be extended to a more complicated case where the substantial uncertainty is imposed on K aircraft. That is, the modified deterministic programming in Eq. (3.16), is solved K times, and the decision variables are updated for each k-th $(= 1, \cdots, K)$ calculation. The decision variables are i) $T_{f^*,r,p_F}(k)$ of the k-th aircraft f^*, ii) $T_{f,r,p_I}(k)$, iii) $T_{f,r,p_T}(k)$, and iv) $T_{f,r,p_F}(k)$ of aircraft f, which follows the k-th aircraft f^*.

Algorithm 3.1 shows the detailed procedure of the proposed heuristic adjustment, which is extended to the complicated case. Suppose that T_{f,r,p_I}, T_{f,r,p_T}, and T_{f,r,p_F} are the static robust solutions determined by the typical deterministic programming in Eq. (3.8) for each aircraft f on route r (line 1 of Algorithm 3.1). If a substantial uncertainty during CDA is imposed on an aircraft f (line 4 of Algorithm 3.1), then the aircraft is referred to as aircraft f^* (line 5 of Algorithm 3.1) and T_{f,r,p_F} of aircraft f^* is adjusted by using Eq. (3.9) (line 10 of Algorithm 3.1). Additionally, the static robust solutions are adjusted by solving Eq. (3.16) (line 13 of Algorithm 3.1). The modified solutions are updated for each k-th $(= 1, \cdots, K)$ calculation (line 14 of Algorithm 3.1). If adjustments are performed for all aircraft for which the flight time during CDA is greater than the extra buffer size, then $T_{f,r,p_I}(k), T_{f,r,p_T}(k)$, and $T_{f,r,p_F}(k)$ for $k = 1, \cdots, K$ can be considered as appropriate scheduling results under dynamic environments; otherwise, the procedure from line 4 to line 15 of Algorithm 3.1 is repeated.

Algorithm 3.1 Heuristic Adjustment

procedure HA(μ^{CDA}, ε, λ, σ^{CDA}, $\tilde{T}^{CDA}_{f,r}, T_{f,r,p_I}, T_{f,r,p_T}, T_{f,r,p_F}$)
  $k \leftarrow 0$
  for each aircraft f on route r **do**
    if $\tilde{T}^{CDA}_{f,r} > \mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}$ **then**
      $f^* \leftarrow f$
      if $k > 0$ **then**
        $T_{f^*,r,p_F} \leftarrow T_{f^*,r,p_F}(k)$
        $T_{f,r,p_I} \leftarrow T_{f,r,p_I}(k)$, $T_{f,r,p_T} \leftarrow T_{f,r,p_T}(k)$, $T_{f,r,p_F} \leftarrow T_{f,r,p_F}(k)$
      end if
      Compute T'_{f^*,r,p_F} using Eq. (3.9)
      $k \leftarrow k + 1$
      $T_{f^*,r,p_F}(k) \leftarrow T'_{f^*,r,p_F}$
      Compute $T'_{f,r,p_I}, T'_{f,r,p_T}, T'_{f,r,p_F}$ for $f = f^* + 1, \cdots, N_F$ using Eq. (3.16)
      $T_{f,r,p_I}(k) \leftarrow T'_{f,r,p_I}$, $T_{f,r,p_T}(k) \leftarrow T'_{f,r,p_T}$, $T_{f,r,p_F}(k) \leftarrow T'_{f,r,p_F}$
    end if
  end for
  return $T_{f,r,p_I}(k), T_{f,r,p_T}(k), T_{f,r,p_F}(k)$ for all aircraft f
end procedure

3.3.2 Sliding Time Window

Suppose that a large number of aircraft arrive through PMS. First, their sequence and schedule can be determined by using the typical deterministic programming in Eq. (3.8), and the obtained result can then be modified under a dynamic environment by using the proposed heuristic adjustment in Section 3.3.1. Because huge computational effort is required to solve the optimization problem, it might be more appropriate to first determine the sequence and schedule of some aircraft approaching PMS in the near future, with subsequent aircraft being added to the problem set as time progresses. Therefore, a sliding time window is introduced to divide the original problem into several small problems.

The operational concept of sliding time window is illustrated in Fig. 3.2, which can be applied as follows. Let us define the length of the time window as 'T^W'. For the first operational interval, the first group of aircraft, with an estimated time of arrival at the initial point, i.e., T^{ETA}_{f,r,p_I}, within $[0, 0 + T^W]$, is included in the optimization problem. For the next operational interval, three different events are considered, as shown in Fig. 3.2. The first event is to add new aircraft (solid triangle in Fig. 3.2) to the problem set, of which T^{ETA}_{f,r,p_I} is within the current time window $[t, t + T^W]$ and was not managed in the previous operational interval. The second event is to eliminate the aircraft (dotted triangle in Fig. 3.2) that have already arrived at the merge point. By eliminating these aircraft from the problem set, the computational effort needed to solve the optimization problem can be maintained at a proper level. The third event is to adjust the sequence and schedule of aircraft (filled triangle at the bottom of Fig. 3.2), which corresponds to the proposed heuristic adjustment in Section 3.3.1. If an aircraft's actual flight time during CDA is greater than

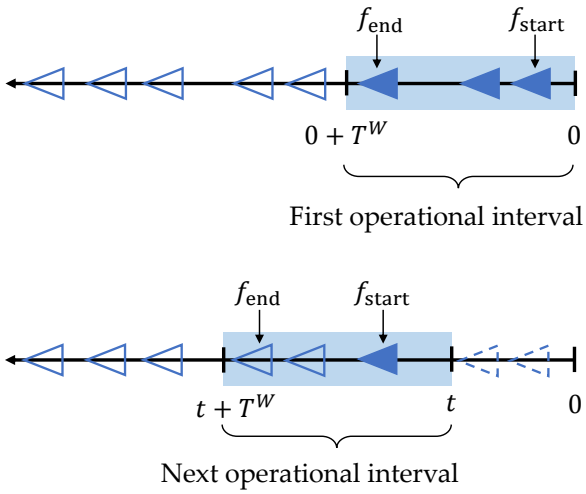


Figure 3.2 Operational concept of sliding time window

the constant value representing the uncertainty of flight time during CDA in the typical deterministic programming, i.e., $\mu^{CDA} + \varepsilon\, \lambda\, \sigma^{CDA}$ in Eq. (3.7), its sequence and schedule as well as those of subsequent aircraft should be modified by solving Eq. (3.16). After the three events are resolved, the sequencing and scheduling of aircraft within the current time window can be determined in a dynamic environment with a low computational load.

3.4 Algorithm Summary

The proposed deterministic programming with algorithm enhancements for dynamic environments is summarized in Algorithm 3.2. When Algorithm 3.2 is started at $t = 0$, the first aircraft is assigned to f_{start}, and the last aircraft in the first group, i.e., the aircraft of which T^{ETA}_{f,r,p_I} is the largest within $[0, 0+T^W]$, which is the first operational interval in Fig. 3.2, is assigned to f_{end} (line 7 of Algorithm 3.2). The parameters f_{start} and f_{end} are used to indicate the earliest and latest aircraft in the current problem set. As previously described, T^E_{f,r,p_I} and T^L_{f,r,p_I} are computed as $0.8T^{ETA}_{f,r,p_I}$ and $1.2T^{ETA}_{f,r,p_I}$, respectively (line 8 of Algorithm 3.2). By solving the typical deterministic programming in Eq. (3.8), the sequences and schedules of the current problem can be calculated (line 9 of Algorithm 3.2). In the next operational interval, as shown in Fig. 3.2, it is determined whether adding aircraft, excluding aircraft, or adjusting the previously determined aircraft schedule is required. If adding aircraft is required, then a new aircraft is added to the problem set, and its schedule is determined by solving the typical deterministic programming in Eq. (3.8) (lines 13–18 of Algorithm 3.2). Note that the parameters T^E_{f,r,p_I} and T^L_{f,r,p_I} of newly added aircraft are then calculated as $t+0.8(T^{ETA}_{f,r,p_I}-t)$ and $t+1.2(T^{ETA}_{f,r,p_I}-t)$, respectively (line 15 of Algorithm 3.2). If deleting aircraft is required, the aircraft referred to as f_{start} is excluded from the problem set (lines 21–26 of Algorithm 3.2). If adjusting the current aircraft schedule is required, then the alternative schedule is determined by solving the modified deterministic programming in Eq. (3.16) (lines 28–40 of Algorithm 3.2). If Algorithm 3.2 is performed in the given time window, i.e., from 0 to $\max(T_{f,r,p_F})$, then $T_{f,r,p_I}(h) \in \mathbb{R}^1, T_{f,r,p_T}(h) \in \mathbb{R}^1$, and $T_{f,r,p_F}(h) \in \mathbb{R}^1$ can be considered as the final scheduling results for the given time window; otherwise, the procedure from line 11 to line 43 is repeated. Because the scheduling results

are updated for each h-th calculation (lines 10, 17, 22–24, and 37–39 of Algorithm 3.2), the solution histories $T_{f,r,p_I}(1,\cdots,h) \in \mathbb{R}^h, T_{f,r,p_T}(1,\cdots,h) \in \mathbb{R}^h$, and $T_{f,r,p_F}(1,\cdots,h) \in \mathbb{R}^h$ can be obtained by using Algorithm 3.2.

Algorithm 3.2 Deterministic Programming for Dynamic Environments

procedure DPDE(F, R, μ^{CDA}, ε, λ, σ^{CDA}, T^W)
  Create $A_{f,r}$, T^{ETA}_{f,r,p_I}, and $\tilde{T}^{CDA}_{f,r}$ for $\forall f \in F$ and $\forall r \in R$
  Create SEP_{f,f',p_I} and SEP_{f,f',p_F}
  for every time step $t = 0, \cdots, \max(T_{f,r,p_F})$ **do**
    if $t = 0$ **then**
      $h \leftarrow 1$
      $f_{\text{start}} \leftarrow 1$, $f_{\text{end}} \leftarrow f$ of which T^{ETA}_{f,r,p_I} is the largest within $[t, t + T^W]$
      $T^E_{f,r,p_I} \leftarrow 0.8\, T^{ETA}_{f,r,p_I}$ and $T^L_{f,r,p_I} \leftarrow 1.2\, T^{ETA}_{f,r,p_I}$ for $f = f_{\text{start}}, \cdots, f_{\text{end}}$
      Compute $(T_{f,r,p_I}, T_{f,r,p_T}, T_{f,r,p_F})$ for $f = f_{\text{start}}, \cdots, f_{\text{end}}$ using Eq. (3.8)
      $T_{f,r,p_I}(h) \leftarrow T_{f,r,p_I}$, $T_{f,r,p_T}(h) \leftarrow T_{f,r,p_T}$, $T_{f,r,p_F}(h) \leftarrow T_{f,r,p_F}$
    else
      for each aircraft $f = f_{\text{end}} + 1, \cdots, N_F$ **do**
        if $T^{ETA}_{f,r,p_I} \leq t + T^W$ **then**
          $h \leftarrow h + 1$, $f_{\text{end}} \leftarrow f_{\text{end}} + 1$
          Update T^E_{f,r,p_I} and T^L_{f,r,p_I} for $f = f_{\text{start}}, \cdots, f_{\text{end}}$
          Compute $(T_{f,r,p_I}, T_{f,r,p_T}, T_{f,r,p_F})$ for $f = f_{\text{start}}, \cdots, f_{\text{end}}$
          $T_{f,r,p_I}(h) \leftarrow T_{f,r,p_I}$, $T_{f,r,p_T}(h) \leftarrow T_{f,r,p_T}$, $T_{f,r,p_F}(h) \leftarrow T_{f,r,p_F}$
        end if
      end for
      for each aircraft $f = f_{\text{start}}, \cdots, f_{\text{end}}$ **do**
        if $T_{f,r,p_F} < t$ **then**
          $h \leftarrow h + 1$, $T_{f,r,p_I}(h) \leftarrow T_{f,r,p_I}(h-1)$ for $f = f_{\text{start}} + 1, \cdots, f_{\text{end}}$
          $T_{f,r,p_T}(h) \leftarrow T_{f,r,p_T}(h-1)$ for $f = f_{\text{start}} + 1, \cdots, f_{\text{end}}$
          $T_{f,r,p_F}(h) \leftarrow T_{f,r,p_F}(h-1)$ for $f = f_{\text{start}} + 1, \cdots, f_{\text{end}}$
          $f_{\text{start}} \leftarrow f_{\text{start}} + 1$
          Update T^E_{f,r,p_I} and T^L_{f,r,p_I} for $f = f_{\text{start}}, \cdots, f_{\text{end}}$
        elseif $t \leq T_{f,r,p_F} \leq t + T^W$
          if $\tilde{T}^{CDA}_{f,r} > \mu^{CDA} + \varepsilon\, \lambda\, \sigma^{CDA}$ **then**
            $T_{f,r,p_I} \leftarrow T_{f,r,p_I}(h)$
            $T_{f,r,p_T} \leftarrow T_{f,r,p_T}(h)$
            $T_{f,r,p_F} \leftarrow T_{f,r,p_F}(h)$
            $f^* \leftarrow f$
            Compute T'_{f^*,r,p_F} using Eq. (3.9)
            $h \leftarrow h + 1$
            $T_{f^*,r,p_F}(k) \leftarrow T'_{f^*,r,p_F}$
            Compute $(T'_{f,r,p_I}, T'_{f,r,p_T}, T'_{f,r,p_F})$ for $f = f^* + 1, \cdots, f_{\text{end}}$
            $T_{f,r,p_I}(h) \leftarrow T'_{f,r,p_I}$
            $T_{f,r,p_T}(h) \leftarrow T'_{f,r,p_T}$
            $T_{f,r,p_F}(h) \leftarrow T'_{f,r,p_F}$
          end if
        end if
      end for
    end if
  end for
  return $T_{f,r,p_I}(h), T_{f,r,p_T}(h), T_{f,r,p_F}(h)$ for all aircraft f
end procedure

3.5 Historical Data Analysis

The uncertainty of flight time during CDA can be identified using historical data analysis for the PMS used at Jeju International Airport. Figure 3.3 shows the air routes, including the PMS, around Jeju International Airport, which is indicated by the small black circle. The Y722 route, indicated by a green dash-dot line, is the busiest air route, by origin-and-destination passenger volume, in the world [80]. The small triangles represent fixes, and the names of some major fixes are indicated. As shown in Fig. 3.3, there exist three PMSs of which availability depends on the wind conditions: two of PMSs (to the left of the airport) are indicated by a blue dotted line, while the other PMS (to the right of the airport) is indicated by a red solid line. In this study, the right PMS is considered as the target PMS, which comprises the initial point DANBI, the merge point HANUL, and the sequencing legs from DANBI to WOOD. This PMS is shared by three inbound traffic flows: DOTOL–PC731–DANBI–WOODO–HANUL (route 1), PC735–DANBI–WOODO–HANUL (route 2), and MAKET–SELIN–WOODO–HANUL (route 3). Note that these routes are vertically separated.

To analyze the arrival traffic flows through the target PMS, historical flight data of a 24-hour period in April 2015 are used. There was a total of 285 arriving flights; among these flights, 113 flights used the target PMS. Among 113 flights, 85.85% used route 1 passing through DOTOL, 4.42% used route 2 passing through PC735, and 9.73% used route 3 passing through MAKET. Four kinds of trajectories are identified as shown in Fig. 3.4: i) good trajectory, ii) unusual trajectory along the sequencing legs, i.e., leg run-offs, iii) unusual trajectory during CDA, and iv) unusual trajectory both along the sequencing legs and during CDA. The number of trajectories per type is summarized in Table 3.1.

Table 3.1 Radar tracking trajectories among 113 aircraft arriving through the target PMS

Trajectory	Good	Unusual along sequencing legs	Unusual during CDA	Both unusual
Number of aircraft	70	30	9	4
Percentage	61.95%	26.55%	7.96%	3.54%

Among 113 flights, only 70 flights (61.95%) arrived at the airport by accurately following the target PMS. Otherwise, when the traffic was heavily congested, a leg run-offs situation, a deviation from CDA, or both resulted. Therefore, it can be concluded that the limitations of the traffic handling capacity of the current PMS are identified, which emphasizes the need for decision support tools for human air traffic controllers in PMS.

Figure 3.5 shows the average flight time consumed in the target PMS and the number of aircraft every 10 minutes during the 24-hour period. It can be seen from Fig. 3.5 that the flight time in the target PMS increases with the number of aircraft, and during the peak hour from 19:50:00 to 20:50:00, a total of 21 aircraft arrived at the airport though the target PMS. The detailed schedules of arriving aircraft during the peak hour are summarized in Table 3.2.

Figure 3.6 shows the histogram of the flight time during CDA for 113 flights. In this study, the uncertainty of flight time during CDA is modelled as a normal distribution, and the mean and standard deviations are 274.60 seconds and 27.75 seconds, respectively. Considering the theoretical value of safe separation between successive aircraft at the merge point, the uncertainty of flight time during CDA could impact the safety of an aircraft, and therefore robust optimization for aircraft sequencing and scheduling in PMS is required to compensate for the uncertainty of flight time during CDA.

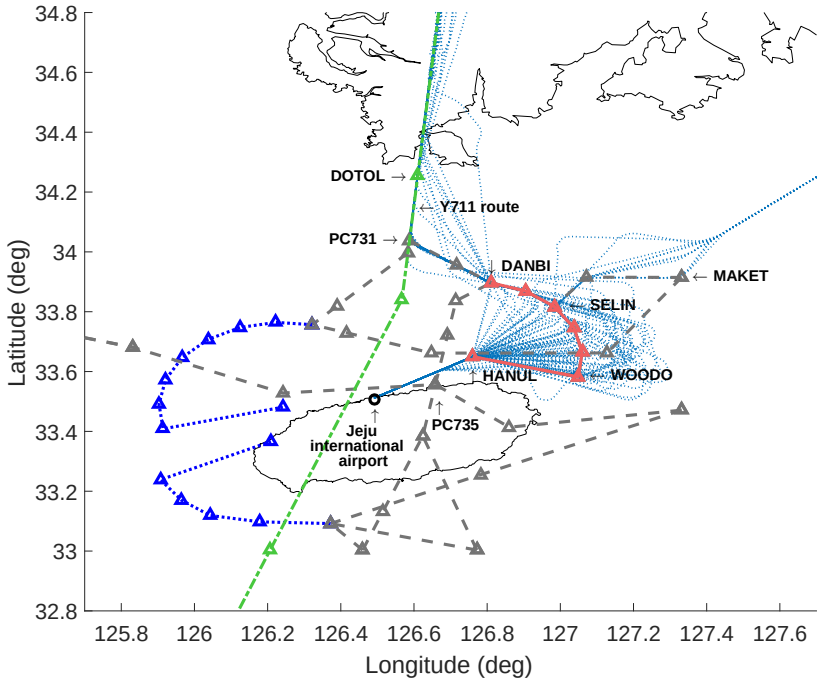
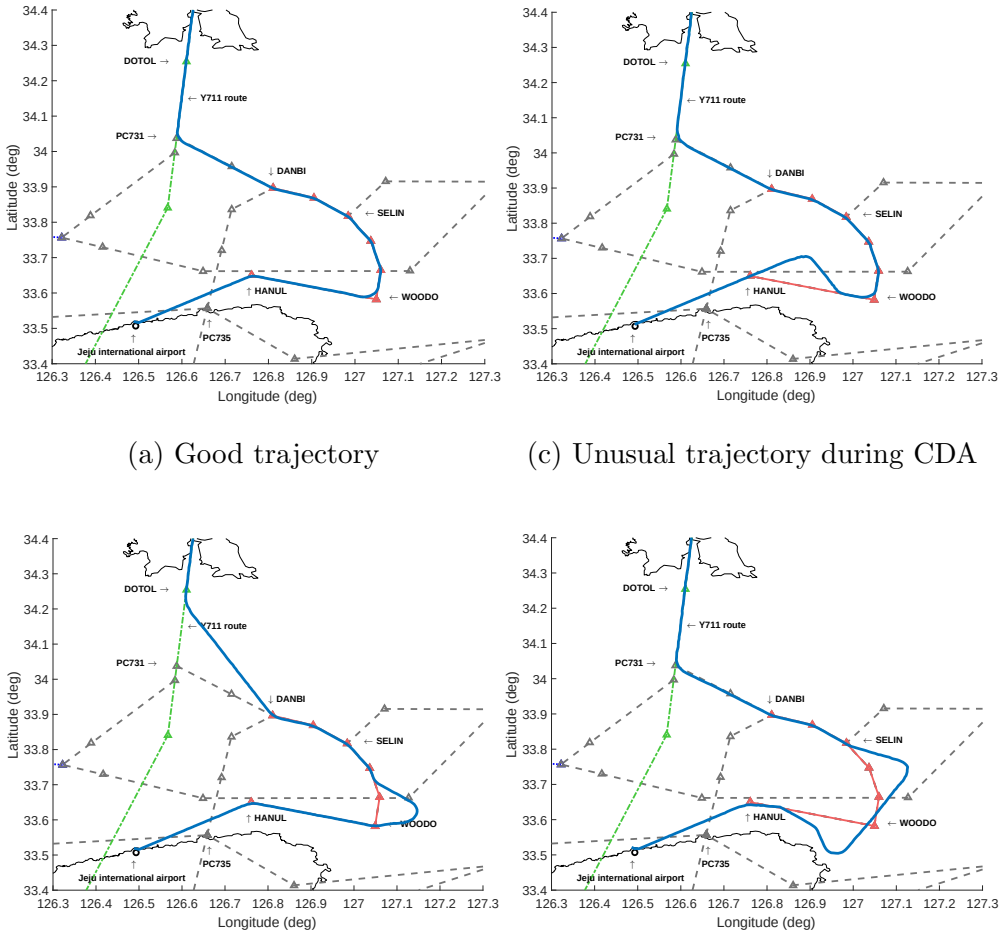


Figure 3.3 PMS configuration of Jeju International Airport and radar tracking trajectories



(a) Good trajectory

(c) Unusual trajectory during CDA

(b) Unusual trajectory along the sequencing leg

(d) Both unusual trajectory

Figure 3.4 Four kinds of radar tracking trajectories

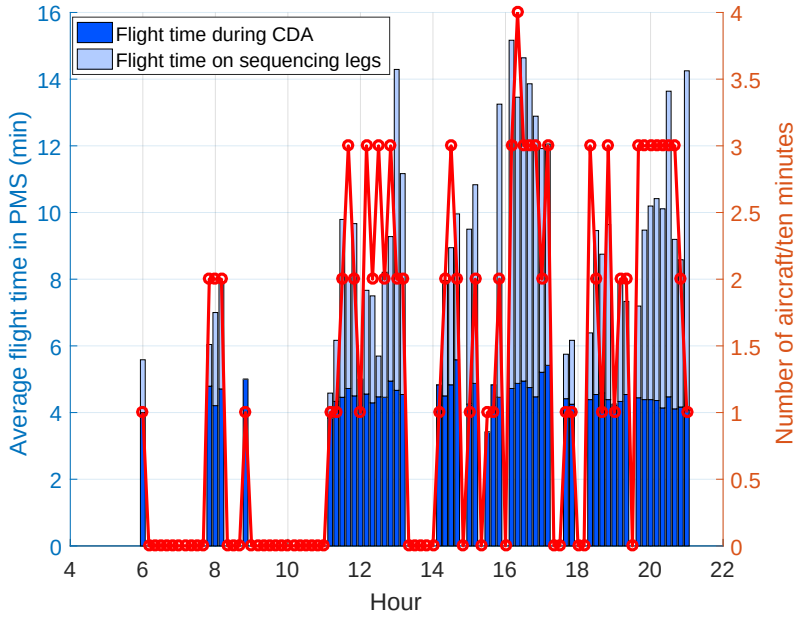


Figure 3.5 Average flight time in PMS (left y-axis) and the number of aircraft arriving through PMS (right y-axis)

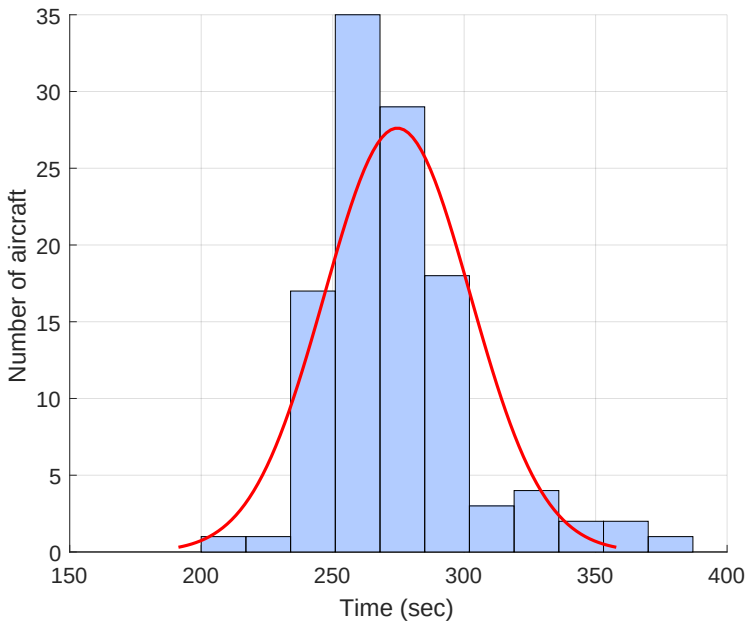


Figure 3.6 Histogram of average flight time during CDA and its distribution, modelled as a normal distribution $N(274.60, 27.75^2)$

Table 3.2 Historical data of arriving aircraft through the target PMS during the peak hour

Aircraft	Route	Class	Arrival Time at p_I (sec)
1	1	Large	0 (19:52:15)
2	1	Large	290 (19:54:15)
3	1	Large	430 (19:58:30)
4	1	Large	1,075 (20:00:45)
5	1	Large	1,260 (20:04:35)
6	1	Large	1,575 (20:07:25)
7	1	Large	1,775 (20:12:55)
8	1	Large	1,920 (20:14:00)
9	1	Large	2,040 (20:14:50)
10	1	Large	2,300 (20:18:30)
11	1	Large	2,415 (20:20:00)
12	3	Large	2,555 (20:26:55)
13	1	Large	2,685 (20:31:35)
14	1	Large	2,995 (20:36:40)
15	1	Large	3,185 (20:40:40)
16	1	Large	3,305 (20:43:40)
17	2	Large	3,585 (20:47:10)
18	1	Large	3,735 (20:47:10)
19	1	Large	4,020 (20:47:10)
20	1	Large	4,285 (20:47:10)
21	1	Large	4,450 (20:47:10)

3.6 Toy Problem

Before performing numerical simulations, a toy problem is introduced to clearly provide intuition about the proposed deterministic programming. For the toy problem, it is considered that three arrival flights exist on two different routes; route 1 is assigned for one flight, and route 2 is assigned for the other flights. The type of flight is arbitrarily assigned among heavy, large, and small. The estimated time of arrival at the initial point is also arbitrarily assigned, but it is considered that the first flights on routes 1 and 2 arrive at their initial points around the same time. In addition, by considering the structure of PMS operated in Jeju International Airport, the flight time during CDA without uncertainty is set to 274.60 seconds. As determined in the historical data analysis, the uncertainty of flight time during CDA is modeled as a normal distribution (i.e., $\tilde{T}^{CDA} \sim N(274.60, 27.75^2)$), and the uncertainty level ε is set to unity. The extra buffer size depends on the reliability level κ, as summarized in Table 3.3. Note that the reliability level represents the probability of the solution to violate the constraints because of the uncertainty [14, 15]. For example, if a 5% reliability level is considered, then the extra buffer size equals to 1.645×27.75 seconds = 45.65 seconds, and the buffer size decreases as the reliability level in-

Table 3.3 Extra buffer sizes depending on the reliability level κ when $\varepsilon = 1$

Reliability level κ (%)	Buffer size (sec)
22.5	20.96
20	23.36
17.5	25.94
15	28.76
12.5	31.92
10	35.56
7.5	39.95
5	45.65
2.5	54.40

creases. In this toy problem, the extra buffer size of 31.92 seconds is considered (i.e., the reliability level is set to 12.5%).

First, let us compare the theoretical results determined by the optimization through MILP in Chapter 2 and the proposed deterministic programming in Chapter 3. Figure 3.7 shows the trajectory of each flight determined by the optimization through MILP. In Fig. 3.7, each triangle denotes the position and orientation of the flight every one minute until each flight reaches the final merge point of PMS. The initial traffic conditions and the theoretical results are summarized in Table 3.4, where $\triangle T_1$ and $\triangle T_2$ are the flight time on the sequencing leg and the flight time during CDA, respectively. On the other hand, Fig. 3.8 and Table 3.5 show the theoretical results when the proposed deterministic programming is implemented. Unlike the optimization result through MILP, the flight time during CDA is set to 306.5 seconds instead of 274.6 seconds to determine robust aircraft sequencing and scheduling under uncertainty in the proposed deterministic programming. Therefore, the total flight time of the proposed deterministic programming is determined as 1,407 seconds which is greater than that of the optimization through MILP (i.e., 1,311 seconds).

Next, suppose that the flight time during CDA of flight 1 is delayed from the theoretical value 274.6 seconds in real operation, and the amount of time delayed is 13.7 seconds. Let us refer this uncertainty as Uncertainty 1. Under Uncertainty 1, the sequencing and scheduling results of the optimization through MILP are shown in Fig. 3.9. As shown in Fig. 3.9, the separation between flights 1 and 3 becomes 96.8 seconds which is less than the minimum safety separation standard, i.e., 110.5 seconds. The detailed values are summarized in Table 3.6. On the other hand, as described in Fig. 3.10 and Table 3.7, the separation between flights 1 and 3 becomes greater than 110.5 seconds by using the proposed deterministic programming, even though Uncertainty 1 is

generated. The reason is that the flight time during CDA (i.e., 306.5 seconds) considered in the theoretical results is greater than the uncertainty of flight time during CDA of flight 1 (i.e., 288.3 seconds).

Lastly, suppose that a substantial uncertainty during CDA is imposed on flight 1; in other words, the flight time during CDA of flight 1 becomes 318.7 seconds in real operation. Let us refer this substantial uncertainty as Uncertainty 2. Note that the theoretical results and the results under Uncertainty 1 could be determined through the typical deterministic programming; in other words, the proposed algorithm enhancements described in Chapter 3.3 are not required. However, because Uncertainty 2 is greater than the constant value, 306.5 seconds, representing the uncertainty of flight time during CDA in the typical deterministic programming, the typical deterministic programming cannot guarantee the minimum safe separation between flights 1 and 3 as shown in Fig. 3.11 and Table 3.8. Therefore, to determine robust sequencing and scheduling under Uncertainty 2, the proposed deterministic programming with a two-level hierarchical architecture is required. Figure 3.12 and Table 3.9 show the toy problem results determined by the proposed deterministic programming. As described in Fig. 3.12 and Table 3.9, by adjusting the arrival time of flight 3 at the final merge point, the separation between flights 1 and 3 becomes greater than 110.5 seconds.

The toy problem results can be summarized as follows. Theoretically, the optimal aircraft sequence and schedule minimizing the total flight time can be achieved by the optimization through MILP. However, when uncertainty is generated in real operation, the optimal aircraft sequence and schedule are not valid. On the other hand, although the total flight time of the proposed deterministic programming increases than that of the optimization through MILP, the aircraft sequencing and scheduling determined by the proposed determin-

istic programming can guarantee the minimum safety separation standard between aircraft under uncertainty.

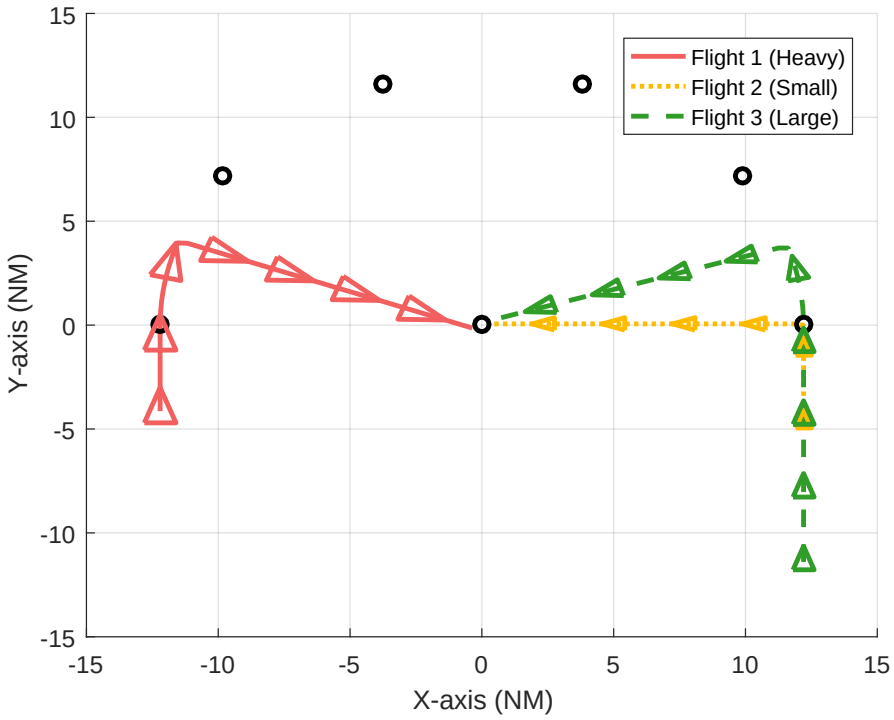


Figure 3.7 Trajectory of each flight theoretically determined by the optimization through MILP

Table 3.4 Theoretical results determined by the optimization through MILP

Initial conditions				Scheduling results (sec)			
Flight	Class	Route	T^{ETA}_{f,r,p_I} (sec)	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	Heavy	1	90.0	72.0	75.5	274.6	422.1
2	Small	2	100.0	80.0	0.0	274.6	354.6
3	Large	2	244.3	195.4	64.6	274.6	532.6

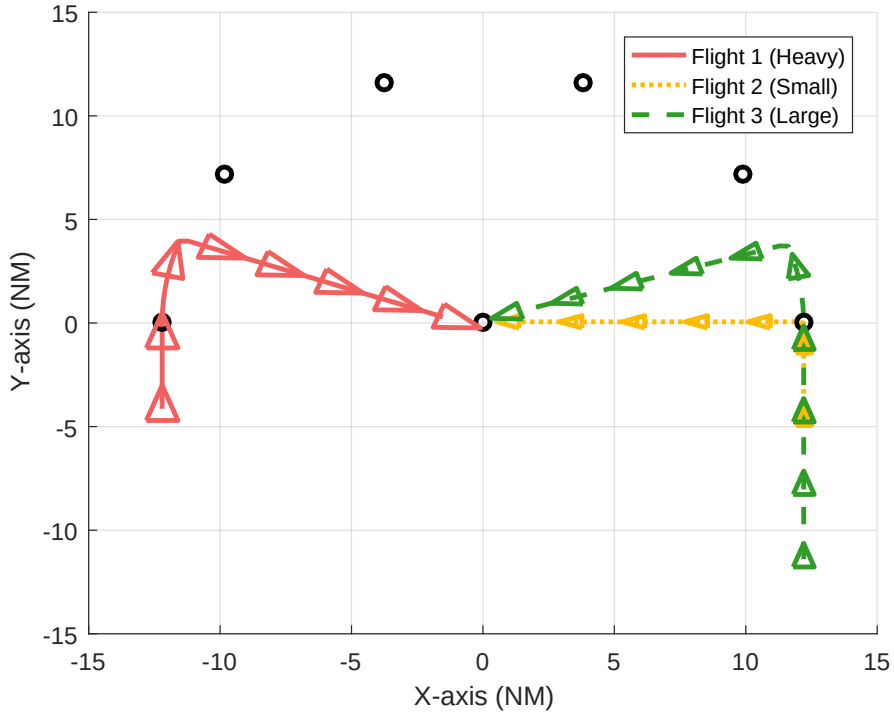


Figure 3.8 Trajectory of each flight theoretically determined by the proposed deterministic programming

Table 3.5 Theoretical results determined by the proposed deterministic programming

Initial conditions				Scheduling results (sec)			
Flight	Class	Route	T^{ETA}_{f,r,p_I} (sec)	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	Heavy	1	90.0	72.0	75.5	306.5	454.0
2	Small	2	100.0	80.0	0.0	306.5	386.5
3	Large	2	244.3	195.4	64.6	306.5	566.5

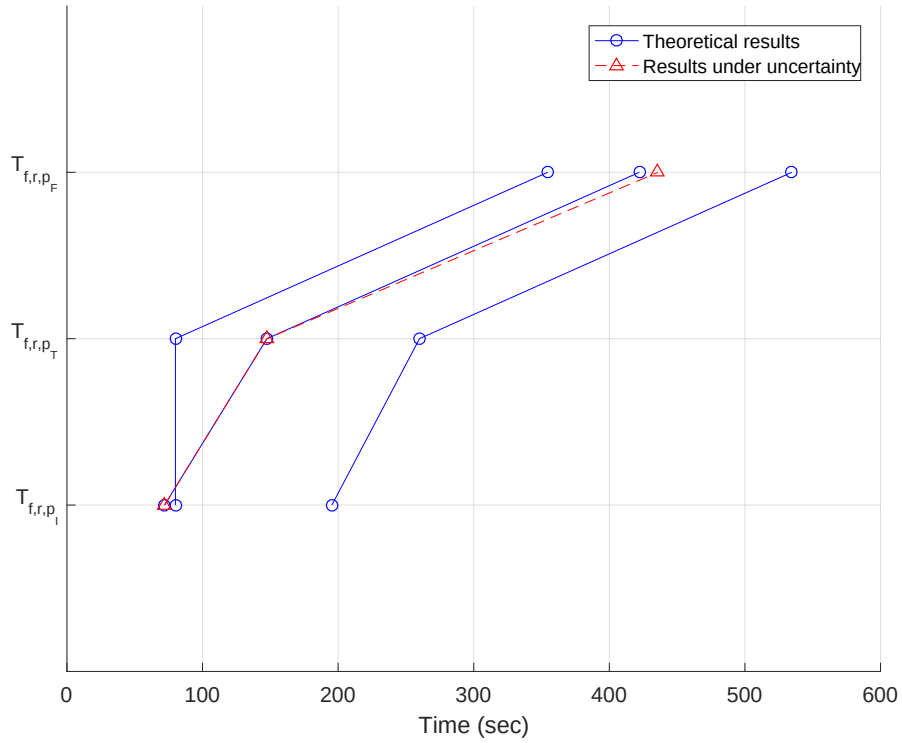


Figure 3.9 Toy problem results under Uncertainty 1 determined by the optimization through MILP

Table 3.6 Toy problem results under Uncertainty 1 determined by the optimization through MILP

Flight	Theoretical results (sec)				Results under uncertainty (sec)			
	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	72.0	75.5	274.6	422.1	72.0	75.5	288.3	435.8
2	80.0	0.0	274.6	354.6	80.0	0.0	274.6	354.6
3	195.4	64.6	274.6	532.6	195.4	64.6	274.6	532.6

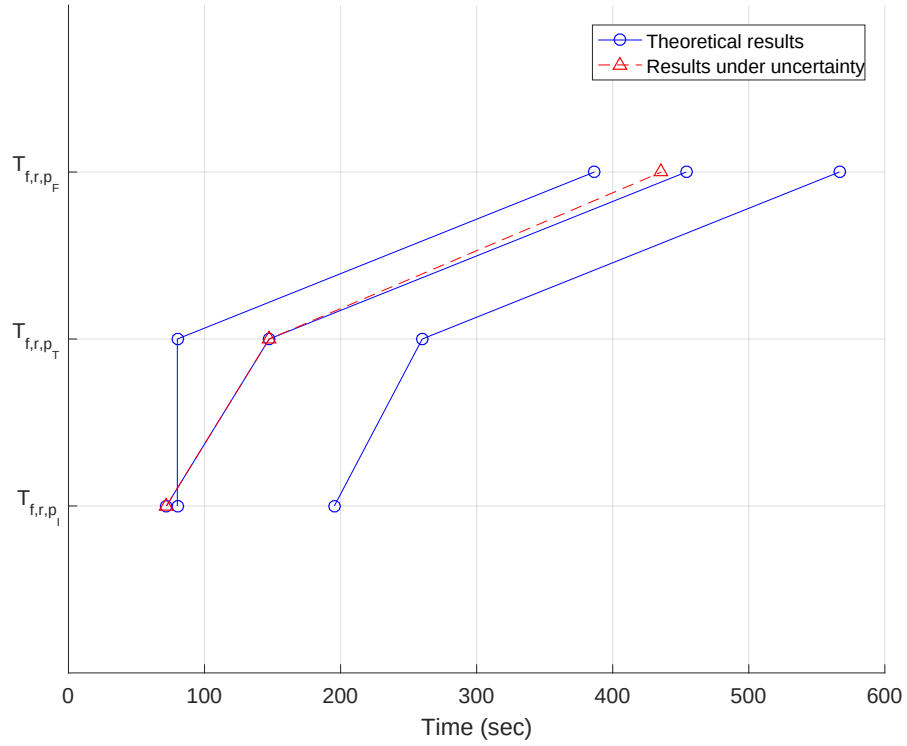


Figure 3.10 Toy problem results under Uncertainty 1 determined by the proposed deterministic programming

Table 3.7 Toy problem results under Uncertainty 1 determined by the proposed deterministic programming

Flight	Theoretical results (sec)				Results under uncertainty (sec)			
	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	72.0	75.5	306.5	454.0	72.0	75.5	288.3	435.8
2	80.0	0.0	306.5	386.5	80.0	0.0	306.5	386.5
3	195.4	64.6	306.5	566.5	195.4	64.6	306.5	566.5

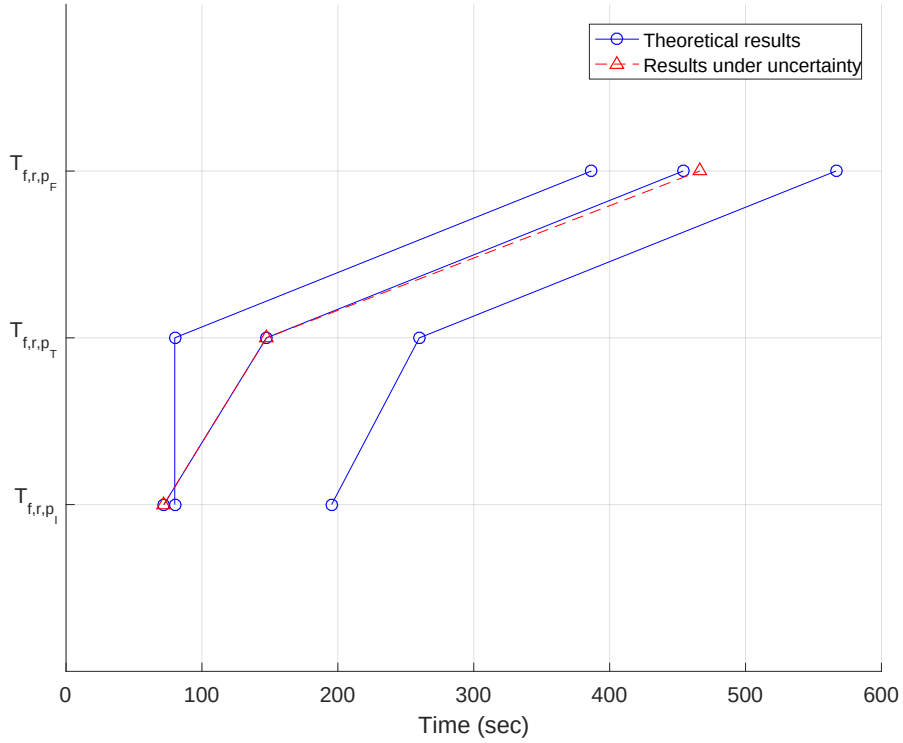


Figure 3.11 Toy problem results under Uncertainty 2 determined by the typical deterministic programming

Table 3.8 Toy problem results under Uncertainty 2 determined by the typical deterministic programming

Flight	Theoretical results (sec)				Results under uncertainty (sec)			
	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	72.0	75.5	306.5	454.0	72.0	75.5	318.7	466.2
2	80.0	0.0	306.5	386.5	80.0	0.0	306.5	386.5
3	195.4	64.6	306.5	566.5	195.4	64.6	306.5	566.5

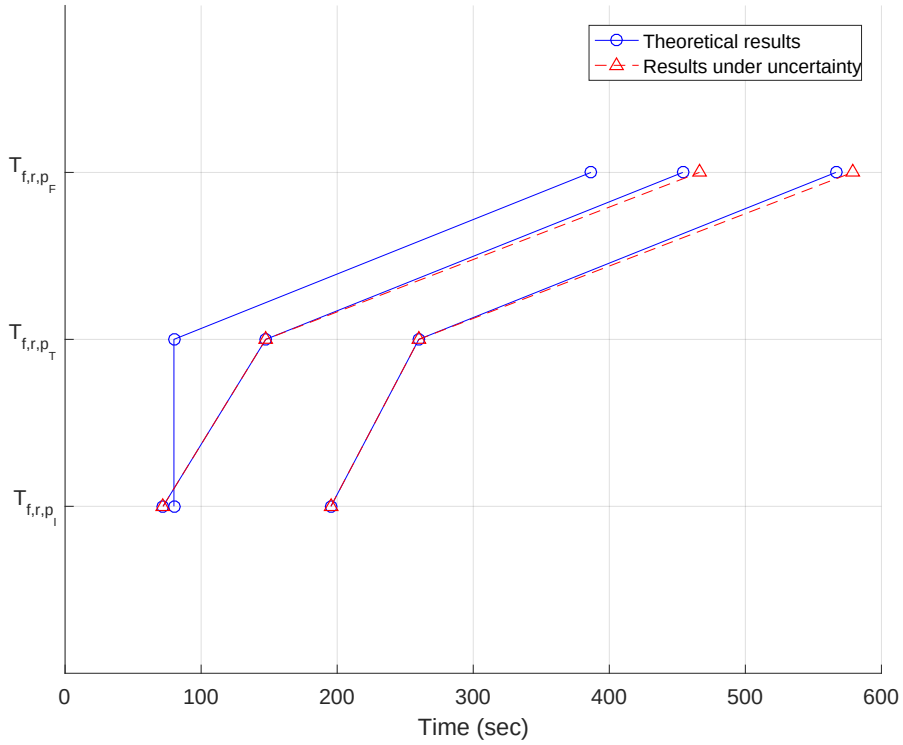


Figure 3.12 Toy problem results under Uncertainty 2 determined by the proposed deterministic programming

Table 3.9 Toy problem results under Uncertainty 2 determined by the proposed deterministic programming

Flight	Theoretical results (sec)				Results under uncertainty (sec)			
	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	72.0	75.5	306.5	454.0	72.0	75.5	318.7	466.2
2	80.0	0.0	306.5	386.5	80.0	0.0	306.5	386.5
3	195.4	64.6	306.5	566.5	195.4	64.6	318.7	578.7

3.7 Numerical Simulation

The proposed deterministic programming in dynamic environments is evaluated based on the real historical data during the peak hour of Jeju International Airport, which are described in Table 3.2. Additionally, to investigate the effect of the extra buffer size on the total flight time and robustness, various extra buffer sizes are considered as summarized in Table 3.3.

The estimated time of arrival at the initial point of PMS is determined based on the real historical data during the peak hour. In a real operation, more time separation at the merge point might be required because of the capacity of the airport runway system and the impact of other arriving and departing aircraft. Therefore, the time separation at the merge point is three times higher than that of theoretical value, e.g., 67.5 seconds $\times$ 3 = 202.5 seconds. The size of the sliding time window is set to be ten minutes.

Figure 3.13 shows the number of adjustments (dashed red line) in the proposed heuristic adjustment and the total flight time (blue solid line) for each buffer size, which is determined based on the reliability level. It is apparent from Fig. 3.13 that more adjustments are required when a smaller buffer size is considered in the typical deterministic programming. The number of adjustments is related to the robustness of the proposed deterministic programming; therefore, the robustness tends to improve as the buffer size increases. However, because more time separation is imposed on each aircraft as the buffer size increases, the total flight time of arriving aircraft through the target PMS increases. This tendency can be identified in Fig. 3.13, where there exists a steady increase in the total flight time from a buffer size of 25.94 seconds to a buffer size of 54.40 seconds. It is interesting to note that the total flight time declines from 20.96 seconds to 25.94 seconds because of the decrease in the number of heuristic

adjustments, which deteriorates the optimality of the solutions. Therefore, the buffer size of 25.94 seconds (i.e., 17.5% reliability level) can be considered as the most appropriate for the proposed deterministic programming in dynamic environments in terms of minimizing the total flight time and the number of interruptions made by human air traffic controllers.

Figure 3.14 and Table 3.10 show the detailed sequencing and scheduling results achieved by the proposed deterministic programming in dynamic environments with the extra buffer when a 17.5% reliability level is considered. As shown in Fig. 3.14, the arrival times at the merge point of the 4th, 10th, and 21st aircraft were delayed with respect to the static robust solutions determined in the typical deterministic programming in Eq. (3.8). In other words, the actual flight time during CDA of these aircraft is greater than the constant value $\mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}$ in the typical deterministic programming when a 17.5% reliability level is considered. To compensate for these delays and to maintain the safe separation between aircraft at the merge point, the proposed heuristic adjustment lengthens the flight time during CDA of the 5th and 11th aircraft and recalculates the turning time of the 12th and 13th aircraft from the sequencing legs, as shown in Fig. 3.14.

Figure 3.15 shows the number of aircraft and the CPU time required for the proposed deterministic programming in dynamic environments for each calculation of the sliding time windows. The aircraft, whose estimated time of arrival are within the range of the first operational interval, i.e., between zero and ten minutes, are scheduled and sequenced in the first calculation. As time progresses, the sliding time window moves forward in time, and a new set of aircraft, with an estimated time of arrival within the current operational interval, is added to the problem set. At the same time, the aircraft arrived at the merge point during the previous operational interval are excluded, and the

sequences and schedules of some aircraft are adjusted when the safe separation between aircraft cannot be maintained because of a substantial uncertainty. As shown in Fig. 3.15, to determine dynamic robust sequences and schedules for 21 aircraft, the proposed deterministic programming is performed 42 times; one is for the first three aircraft at the first operational interval, 18 times are performed to add an aircraft whenever a new aircraft enters to the current operational window, 3 times are to adjust the schedule and sequence of aircraft when a substantial uncertainty is generated during CDA, and 20 times are to eliminate the aircraft which arrives at the merge point of PMS. Specifically, at the third calculation where the operational window is from 476 seconds to 1,076 seconds, the fourth aircraft is added to the problem set. The decision variables T_{f,r,p_I}, T_{f,r,p_T}, and T_{f,r,p_F} for the fourth aircraft are then determined as 955 seconds, 955 seconds, and 1,256 seconds, respectively. And then, at the fourth calculation where the operational window is from 536 seconds to 1,136 seconds, the second aircraft is excluded from the problem set. At the fifth calculation where the operational window is from 661 seconds to 1,261 seconds, the decision variables T_{f,r,p_I}, T_{f,r,p_T}, and T_{f,r,p_F} of the fifth aircraft are determined as 1,140 seconds, 1,158 seconds, and 1,459 seconds, respectively. However, during the seventh calculation, the fourth and fifth aircraft are rescheduled because the actual flight time during CDA of the fourth aircraft is larger than the predefined extra buffer size. Therefore, T_{f,r,p_F} of the fourth aircraft is changed from 1,256 seconds to 1,280 seconds. Also, the decision variable T_{f,r,p_F} of the fifth aircraft is updated as 1,483 seconds, respectively.

In Table 3.11, the minimum, mean, and maximum number of aircraft considered in each calculation and the minimum, mean, and maximum CPU time required to solve the problems are summarized. As shown in Table 3.11, by using the sliding time window, the number of aircraft included in each problem

set is maintained less than six. Therefore, the proposed deterministic programming for dynamic environments can be solved efficiently and implemented over a long time period (approximately one hour) with a low computational load. Note that the computation is performed using MATLAB [81] and a desktop PC with an Intel Core (3.60 GHz) processor.

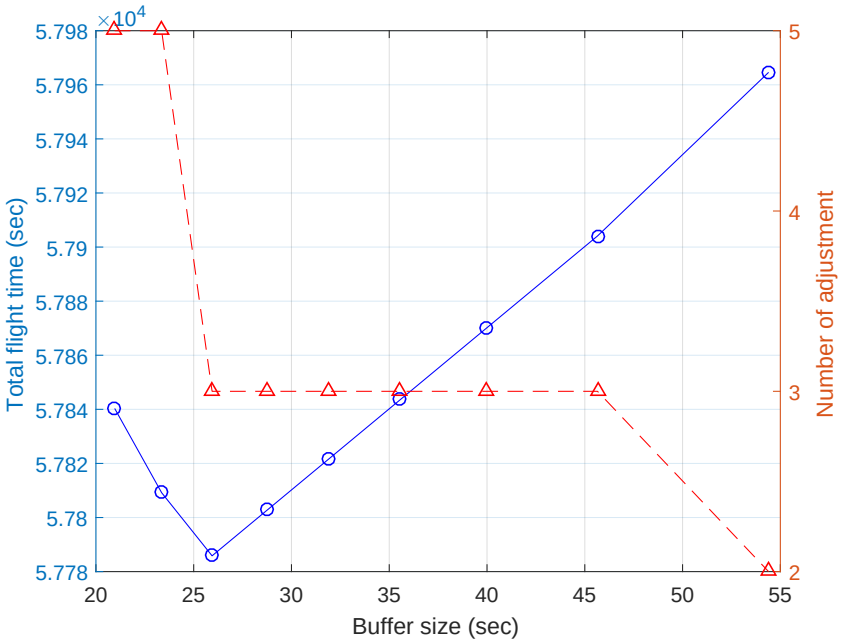


Figure 3.13 The number of heuristic adjustments and the total flight time for each buffer size

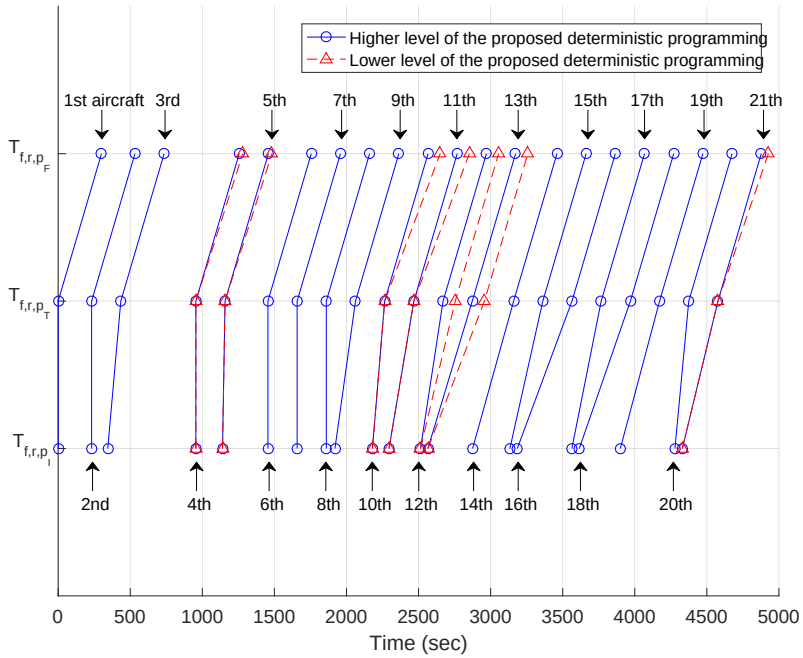


Figure 3.14 Sequencing and scheduling results for the extra buffer size with a 17.5% reliability level

Table 3.10 Detailed sequencing and scheduling results for the extra buffer size with a 17.5% reliability level

Aircraft	T_{f,r,p_I} (sec)	T_{f,r,p_T} (sec)	$\tilde{T}_{f,r}^{CDA}$ (sec)	T_{f,r,p_F} (sec)
1	0	0	301	301
2	232	232	301	533
3	344	435	301	735
4	955	955	325	1,280
5	1,140	1,158	325	1,483
6	1,455	1,455	301	1,756
7	1,658	1,658	301	1,958
8	1,860	1,860	301	2,161
9	1,920	2,063	301	2,363
10	2,180	2,265	385	2,650
11	2,295	2,468	385	2,852
12	2,514	2,755	301	3,055
13	2,565	2,957	301	3,258
14	2,875	3,160	301	3,460
15	3,134	3,362	301	3,663
16	3,185	3,565	301	3,865
17	3,564	3,767	301	4,068
18	3,615	3,970	301	4,270
19	3,900	4,172	301	4,473
20	4,279	4,375	301	4,675
21	4,330	4,577	350	4,927

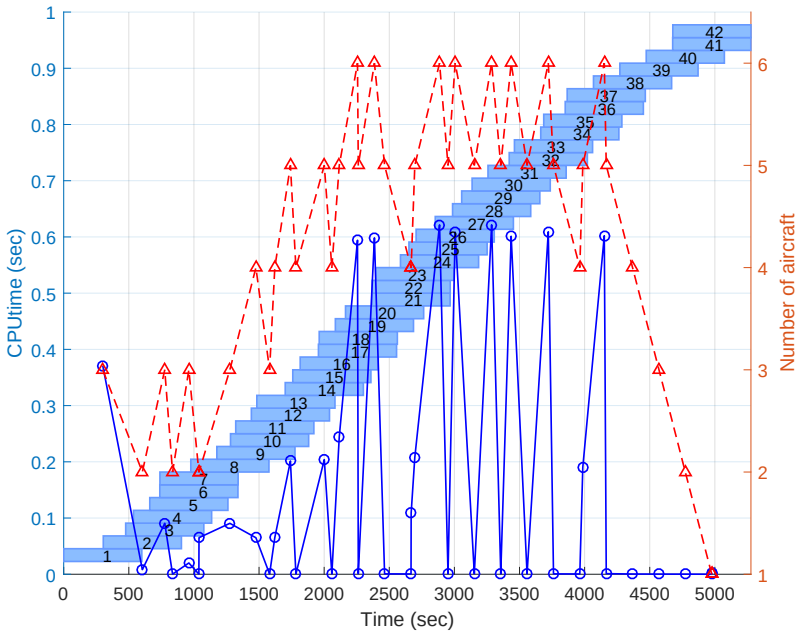


Figure 3.15 The number of aircraft and the CPU time required for each calculation for the extra buffer size with a 17.5% reliability level

Table 3.11 The minimum, mean, and maximum number of aircraft and the CPU time required for each calculation for the extra buffer size with a 17.5% reliability level

The number of aircraft	Minimum	1
	Mean	4.17
	Maximum	6
CPU time (sec)	Minimum	0.00
	Mean	0.16
	Maximum	0.62

Chapter 4

Stochastic Programming for Aircraft Arrival Sequencing and Scheduling under Uncertainty

This chapter describes robust optimization based on stochastic programming for the aircraft sequencing and scheduling problem in PMS to compensate for the uncertainty of flight time during CDA. The general formulation for two-stage stochastic programming is described in Section 4.1. Two existing stochastic approaches are briefly summarized in Section 4.1: i) deterministic equivalent programming in Section 4.1.1 and ii) two-stage stochastic programming based on GA in Section 4.1.2. Then, two-stage stochastic programming based on PSO is proposed with a detailed formulation for the first and second stage decisions in Section 4.2.1. In Section 4.2.2, the aircraft sequencing problem is newly represented based on the random key representation to determine the first stage decision by using PSO. In Section 4.3, to clearly understand the proposed two-stage stochastic programming based on PSO, the toy problem is solved. By performing three different numerical simulations in Section 4.4, the appropriate number of scenarios required in the second stage is determined and the performance of the proposed two-stage stochastic programming based on PSO is compared with that of the deterministic programming proposed in Chapter 3 and other stochastic approaches.

4.1 Two-Stage Stochastic Programming

Let us reconsider the general MILP problem described in Eq. (3.2). By incorporating the uncertainty existing in the right-hand-side parameter of the inequality constraint in Eq. (3.2), two-stage stochastic programming can be represented as follows:

$$\min_{x,y} \quad c^T x + \mathbb{E}_\xi \left[\phi(x, y, \xi)\right] \tag{4.1}$$

subject to

$$Ax \leq b, \quad x \in X \tag{4.2}$$

where the symbol $\mathbb{E}_\xi$ denotes expectation with respect to the distribution of random variable ξ, and the function $\phi(x, y, \xi)$ is given as follows:

$$\phi(x, y, \xi) = \min_y q(\xi)^T y \tag{4.3}$$

subject to

$$Tx + Wy \leq h(\xi), \quad y \in Y \tag{4.4}$$

If the uncertainty is considered to be a finite set of scenarios, i.e., $\omega \in \Omega, \Omega = \{1, \cdots, N_\Omega\}$, Eqs. (4.1)–(4.4) can be re-established as follows [60, 61]:

$$\min_{x,y_\omega} \quad c^T x + \sum_{\omega=1}^{N_\Omega} \pi_\omega q_\omega^T y_\omega \tag{4.5}$$

subject to

$$Ax \leq b, \tag{4.6}$$

$$Tx + Wy_\omega \leq h_\omega, \quad \forall \omega = 1, \cdots, N_\Omega \tag{4.7}$$

where the variables to be optimized are divided into the first stage decision variables x and the second stage decision variables y_ω. Note that the decision

in the first stage does not depend on each scenario ω, and Eq. (4.6) is the constraint for the first stage decision. On the other hand, the decision in the second stage is associated with a particular realization of h_ω in Eq. (4.7). The objective function of two-stage stochastic programming in Eq. (4.5) is composed of the first stage decision and the expected cost of the second stage decision. In addition, π_ω represents the probability of each scenario ω, and c and q_ω are weighting parameters for the first and second stage decisions, respectively.

4.1.1 Deterministic Equivalent Programming (DEP)

To determine robust solutions of two-stage stochastic programming in Eqs. (4.5)–(4.7), the first approach is to derive the DEP problem based on scenario decomposition [68]. By introducing x_ω according to the number of scenarios and the so-called nonanticipativity constraints such that $x_1 = x_2 = \cdots = x_{N_\Omega}$, Eqs. (4.5)–(4.7) can be represented as follows:

$$\min_{x,y_\omega} \sum_{\omega=1}^{N_\Omega} \pi_\omega \Big\{ c^T x_\omega + q_\omega^T y_\omega \Big\} \tag{4.8}$$

subject to

$$Ax_\omega \leq b, \ \ \forall \omega = 1, \cdots, N_\Omega \tag{4.9}$$

$$Tx + Wy_\omega \leq h_\omega, \ \ \forall \omega = 1, \cdots, N_\Omega \tag{4.10}$$

$$x_\omega - x_{\omega+1} = 0, \ \ \forall \omega = 1, \cdots, N_\Omega - 1 \tag{4.11}$$

Because the DEP problem presented in Eqs. (4.8)–(4.11) belongs to the class of MILP, a standard optimization software for MILP such as CPLEX [82] can be utilized to solve it. However, the number of second stage decisions y_ω and the number of nonanticipativity constraints in Eq. (4.11) grow linearly with the number of scenarios; therefore, the DEP problem with a large number of variables and constraints requires a large amount of computational time.

4.1.2 Two-Stage Stochastic Programming based on GA

An alternative approach is the two-stage stochastic programming based on an evolutionary algorithm, especially GA [22, 23]. Equations (4.5)–(4.7) can be divided into a master problem and sub-problems. The master problem is a function of the first stage decision x only. Therefore, it can be represented as follows:

$$\min_{x} \quad c^T x + \sum_{\omega=1}^{N_\Omega} \pi_\omega Q_\omega(x) \tag{4.12}$$

subject to

$$Ax \leq b \tag{4.13}$$

For a given x, the sub-problems can be set up as follows:

$$Q_\omega(x) = \min_{y_\omega} \; q_\omega^T y_\omega \tag{4.14}$$

subject to

$$Tx + Wy_\omega \leq h_\omega, \quad \forall \omega = 1, \cdots, N_\Omega \tag{4.15}$$

In [22,23], the master problem presented in Eqs. (4.12)–(4.13) is solved by using GA. Each individual in GA represents the first stage decision x and evaluates its fitness as described in Eqs. (4.12)–(4.13). Note that to calculate the fitness, the sub-problems of Eqs. (4.14)–(4.15) are solved by using CPLEX. By performing the GA operations, i.e., the selection, crossover, and mutation operations, the offspring of the current generation can provide better solutions than those of the previous generation. Eventually, the optimal first stage decision x resulting in the minimum fitness for N_Ω scenarios can be determined.

4.2 Two-Stage Stochastic Programming based on PSO

4.2.1 Master and Sub-Problems

Before applying the two-stage stochastic programming based on PSO, let us reconsider the uncertain inequality constraint in Eq. (3.1). As described in Eq. (3.1), the uncertainty arising in the CDA trajectory is considered in this study. The uncertainty of flight time during CDA $\tilde{T}_{f,r}^{CDA}$ in Eq. (3.1) is closely related to the continuous decision variable $T_{f,r,p}$; otherwise, it does not connect with the binary decision variable $S_{f,f',r,r',p}$. In other words, the continuous decision variable $T_{f,r,p}$ is associated with each realization of the uncertainty of flight time during CDA $\tilde{T}_{f,r}^{CDA}$, while the binary decision variable $S_{f,f',r,r',p}$ is not. For this reason, the first stage decision x is defined as the binary decision variable $S_{f,f',r,r',p}$ for the aircraft sequencing problem, and the second stage decision y_ω is defined as the continuous decision variable $T_{f,r,p}$ for the aircraft scheduling problem.

The master problem for the aircraft arrival sequencing and scheduling problem in PMS can be represented as follows:

$$\min_{S_{f,f',r,r',p}} \frac{1}{N_\Omega} \sum_{\omega=1}^{N_\Omega} Q_\omega(S_{f,f',r,r',p}) \tag{4.16}$$

subject to Eqs. (2.4) and (2.5). Note that, in Eq. (4.16), it is assumed that each scenario ω has identical probability with $1/N_\Omega$. In addition, for a given $S_{f,f',r,r',p}$, the resulting sub-problems for the aircraft arrival sequencing and scheduling problem in PMS can be written as follows:

$$Q_\omega(S_{f,f',r,r',p}) \quad = \quad \min_{T_{f,r,p}} \sum_{f=1}^{N_F} \sum_{r=1}^{N_R} A_{f,r} T_{f,r,p_F} \tag{4.17}$$

subject to Eqs. (2.2)–(2.3), Eqs. (2.6)–(2.8), and Eq. (3.1). Note that for each scenario ω, the uncertainty of flight time during CDA, i.e., $\tilde{T}_{f,r}^{CDA}$ in Eq. (3.1),

is sampled from the normal distribution function, which is identified through the historical data analysis performed in Section 3.5.

4.2.2 Random Key Representation

To solve the master problem described in Eq. (4.16) by using PSO, the random key representation [75, 76] is utilized in this study. In other words, the position vector $x_s \in \mathbb{R}^{N_F \cdot (N_P - 1)}$ of the particle s ($s \in S$, $S = \{1, \cdots, N_S\}$), where N_S is the swarm size, is defined as follows:

$$x_s = [x_{sp_I} \;\; x_{sp_F}] \tag{4.18}$$

$$x_{sp_I} = [x_{sp_I 1} \;\; x_{sp_I 2} \;\; \cdots \;\; x_{sp_I N_F}] \tag{4.19}$$

$$x_{sp_F} = [x_{sp_F 1} \;\; x_{sp_F 2} \;\; \cdots \;\; x_{sp_F N_F}] \tag{4.20}$$

where x_{sp_I} and x_{sp_F} represent the continuous value for all aircraft $f \in F$ at the initial and merge points, respectively. For example, the element of the position vector $x_{sp_I} = [0.25 \;\; 0.43 \;\; 0.06 \;\; 0.85 \;\; 0.17]$ represents a five aircraft sequence, such as 3→5→1→2→4, at the initial point p_I. By using this solution representation, the operational constraint for the aircraft sequencing problem described in Eqs. (2.4) and (2.5) can be automatically satisfied.

The detailed procedure for transforming the element of the position vector x_s of particle s, i.e., x_{sp_I} and x_{sp_F}, to the binary decision variable $S_{f,f',r,r',p}$ for the aircraft sequencing problem at a point p is summarized in Algorithm 4.1. Note that the output of Algorithm 4.1 is the matrix $S2$ which contains each binary decision variable $S_{f,f',r,r',p}$ for aircraft sequencing problem at a point p for $\forall f \in F, \forall f' \in F, f \neq f', \forall r \in R, \forall r' \in R$.

Algorithm 4.1 Random Key Representation

```
1: procedure RKR(x_sp, F, R, A_{f,r})
2:     for i from 1 to N_F do
3:         S0(i, 1) ← x_sp(1, i)
4:         S0(i, 2) ← i
5:     end for
6:     [ign, seq] ← sort(S0(:, 1))
7:     S1 ← zeros(1, N_F^2)
8:     for i from 1 to N_F do
9:         for j from 1 to N_F do
10:            if find(seq== i) < find(seq== j) then
11:                S1(1, (i − 1) · N_F + j ) ← 1
12:                S1(1, (j − 1) · N_F + i ) ← 0
13:            else
14:                S1(1, (i − 1) · N_F + j ) ← 0
15:                S1(1, (j − 1) · N_F + i ) ← 1
16:            end if
17:        end for
18:    end for
19:    k ← 0
20:    S2 ← zeros(1, N_F^2 · N_R^2)
21:    for i from 1 to N_F do
22:        for j from 1 to N_F do
23:            k ← k + 1
24:            if S1(1, k) == 1 then
25:                if A_{f,r} == 1 when f ← i, r ← r_L then
26:                    r1 ← 1
27:                else
28:                    r1 ← 2
29:                end if
30:                if A_{f,r} == 1 when f ← j, r ← r_L then
31:                    r2 ← 1
32:                else
33:                    r2 ← 2
34:                end if
35:                S2(1, (i − 1) · N_F · N_R^2 + (j − 1) · N_R^2 + (r1 − 1) · N_R + r2) ← 1
36:            end if
37:        end for
38:    end for
39:    return S2
40: end procedure
```

4.2.3 Algorithm Summary

With the new solution representation in Eqs. (4.18)–(4.20), a fitness J_s^{iter} at each iteration can be defined as follows:

$$J_s^{iter} = \frac{1}{N_\Omega} \sum_{\omega=1}^{N_\Omega} Q_\omega(S_{f,f',r,r',p}) \tag{4.21}$$

where $Q_\omega(S_{f,f',r,r',p})$ can be determined by solving the sub-problems of Eq. (4.17). To determine the optimal aircraft sequence that minimizes Eq. (4.21), the velocity vector $u_s \in \mathbb{R}^{N_F \cdot (N_P - 1)}$ and position vector x_s of the particle s are generated using a random uniform distribution within bounds at the beginning. Then, updates are performed at each iteration as follows:

$$u_s^{iter} = W u_s^{iter-1} + c_1 r_1 (x_{s_p}^* - x_s^{iter-1}) + c_2 r_2 (x_g^* - x_s^{iter-1}) \tag{4.22}$$

$$x_s^{iter} = x_s^{iter-1} + u_s^{iter} \tag{4.23}$$

where $x_{s_p}^* \in \mathbb{R}^{N_F \cdot (N_P - 1)}$ and $x_g^* \in \mathbb{R}^{N_F \cdot (N_P - 1)}$ are the stored best previous position of the particle s and the position of the best particle among all of the particles, respectively, c_1 and c_2 are the acceleration constants, and r_1 and r_2 are uniform random values between 0 and 1. In Eq. (4.22), the second and third terms represent the personal and social experiences, respectively. Note that one key strength of PSO is to make a balance between the personal and social experiences. If the term representing the personal experience is larger than the social experience, then a particle remains its current position; otherwise, the particle updates its velocity based on the social experience. In addition, W is an inertia weight, whose role is crucial for the PSO's convergence behavior. A high inertia weight means that the particles tend to maintain the current direction. On the other hand, a low inertia weight means that the particles tend to follow their personal and social experiences [83]. In the exisitng works on PSO, the inertia weight was set to a constant value [84], and a time-decreasing inertia

weight value was used [85]. Note that, the acceleration constants and the inertia weight should be carefully determined, because they provide balance between the global and local exploration ability of the swarm, which results in better convergence performance.

Algorithm 4.2 shows the detailed procedure of the proposed two-stage stochastic programming based on PSO. Each particle's position x_s and velocity u_s are initialized with a uniform distribution in the specified range [0 4] and [-1,000 1,000] (default), respectively (line 2 of Algorithm 4.2). After all particles' fitness values are computed by using Eq. (4.21), the values are compared with each other, and then the best of the personal bests $x^*_{s_p}$ is set to the global best x^*_g (line 5 of Algorithm 4.2). Each particle updates its position x_s and velocity u_s using Eqs. (4.22) and (4.23), thereby they move closer to the optimal solution with the personal best $x^*_{s_p}$ and the global best x^*_g (lines 6–7 of Algorithm 4.2). Each particle evaluates its current position x_s by computing its fitness value J_s^{iter} based on Eq. (4.21) (line 9 of Algorithm 4.2). If the current fitness value of a particle J_s^{iter} is less than its previous fitness value J_s^{iter-1}, then it updates its personal best $x^*_{s_p}$ (line 11 of Algorithm 4.2). If one of the stopping criteria is met, then the global best x^*_g is returned as the optimal solution; otherwise, the procedure from line 5 to line 17 of Algorithm 4.2 is repeated.

Algorithm 4.2 Two-Stage Stochastic Programming based on PSO

procedure TSSPPSO(F, S)
  Initialize x_s and u_s for all particles $s \in S$
  Initialize the $x^*_{s_p}$ and the cost J^0_s such as $x^*_{s_p} \leftarrow x_s$ and $J^0_s \leftarrow \infty$
  while stopping criteria are not satisfied **do**
    Find $x^*_{s_p}$ and set the best of $x^*_{s_p}$ as x^*_g by using Eq. (4.21)
    Update each particle's velocity by using Eq. (4.22)
    Update each particle's position by using Eq. (4.23)
    for each $s = 1, \cdots, N_S$ **do**
      Compute J^{iter}_s by using Eq. (4.21)
      if $J^{iter}_s < J^{iter-1}_s$ **then**
        $x^*_{s_p} \leftarrow x_s$
      else
        $J^{iter}_s \leftarrow J^{iter-1}_s$
      end if
    end for
    Check the stopping criteria
    $iter \leftarrow iter + 1$
  end while
  return $x^* \leftarrow x^*_g$
end procedure

4.3 Toy Problem

In this section, the toy problem introduced in Section 3.6 is solved by using the proposed two-stage stochastic programming based on PSO. For simplicity, the number of scenarios required in the second stage is set to 10. For 10 different realizations of uncertainty, the optimal aircraft sequence determined by the proposed two-stage stochastic programming based on PSO is as follows: 1→2→3 at the initial point and 2→1→3 at the merge point. Let us consider the ordered list which contains the sum of 10 different realizations of uncertainty. In the ordered list, the worst uncertainty imposed on flights 1, 2, and 3 (i.e., the 100th percentiles of this list) are 306.3 seconds, 292.2 seconds, and 348.9 seconds, respectively. When the worst uncertainty is considered, the aircraft sequencing and scheduling results determined by the proposed two-stage stochastic programming based on PSO are illustrated in Fig. 4.1 and Table 4.1. Note that the total flight time of the proposed two-stage stochastic programming based on PSO is 1,389 seconds, which is less than that of the deterministic programming, i.e., 1,407 seconds.

Suppose that Uncertainty 2, which was introduced in Chapter 3.6, is generated. Under Uncertainty 2, the aircraft sequencing and scheduling results determined by the two-stage stochastic programming based on PSO are given in Fig. 4.2 and Table 4.2. As described in Fig. 4.2 and Table 4.2, the minimum safe separation between flights 1 and 3 can be successfully maintained, even though Uncertainty 2 is generated. Furthermore, the total flight time of the two-stage stochastic programming based on PSO is 1,389 seconds which is less than that of the deterministic programming, i.e., 1,431 seconds.

Through this toy problem, the following characteristics and performance of the deterministic programming and the two-stage stochastic programming

based on PSO can be identified. First, the proposed deterministic programming is quite simple and results in computational benefits, but provides conservative robust solutions by replacing an uncertain parameter with a constant value which becomes larger as the robustness increases. For this reason, the proposed deterministic programming has a tendency to increase the total flight time. On the other hand, although the computational load of the proposed two-stage stochastic programming based on PSO is quite high, it provides less conservative robust solutions than the proposed deterministic programming by considering an uncertain parameter having a probability distribution function. In other words, the proposed two-stage stochastic programming based on PSO has an advantage in terms of the total flight time.

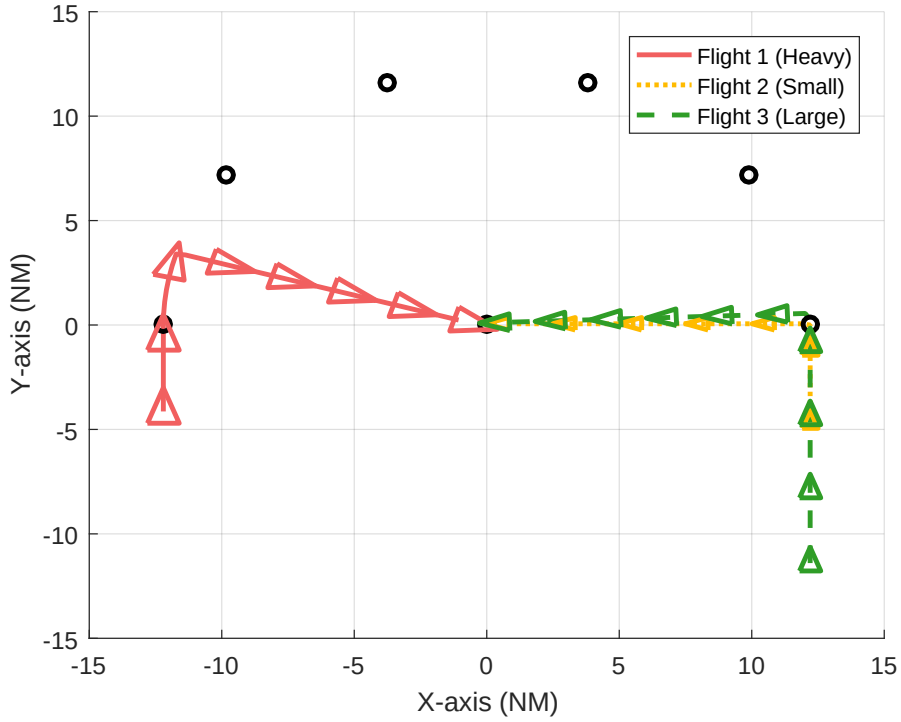


Figure 4.1 Trajectory of each flight theoretically determined by the proposed two-stage stochastic programming based on PSO

Table 4.1 Theoretical results determined by the proposed two-stage stochastic programming based on PSO

Initial conditions				Scheduling results (sec)			
Flight	Class	Route	T^{ETA}_{f,r,p_I} (sec)	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	Heavy	1	90.0	72.0	61.4	306.3	439.7
2	Small	2	100.0	80.0	0.0	292.2	372.2
3	Large	2	244.3	195.4	7.8	348.9	552.2

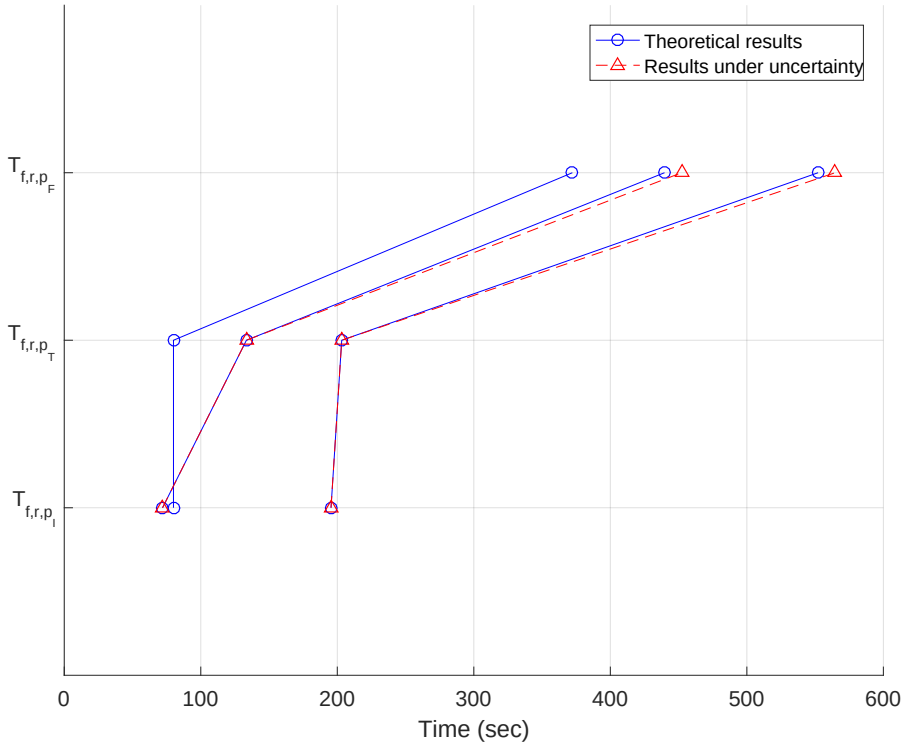


Figure 4.2 Toy problem results under Uncertainty 2 determined by the proposed two-stage stochastic programming based on PSO

Table 4.2 Toy problem results under Uncertainty 2 determined by the proposed two-stage stochastic programming based on PSO

Flight	Theoretical results (sec)				Results under uncertainty (sec)			
	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}	T_{f,r,p_I}	$\triangle T_1$	$\triangle T_2$	T_{f,r,p_F}
1	72.0	61.4	306.5	439.7	72.0	61.4	318.7	452.1
2	80.0	0.0	292.2	372.2	80.0	0.0	292.2	372.2
3	195.4	7.8	348.9	552.2	195.4	7.8	361.4	564.6

4.4 Numerical Simulation

The performance of the proposed two-stage stochastic programming based on PSO is demonstrated through three different numerical simulations. In the first simulation, because the number of scenarios is closely related to the accuracy and the computational load, the appropriate number of scenarios required in the second stage of the proposed two-stage stochastic programming based on PSO is investigated. In the second simulation, the simulation results of the proposed two-stage stochastic programming based on PSO are compared with those of the proposed deterministic programming in Chapter 3. Lastly, the performance of the proposed two-stage stochastic programming based on PSO is compared with that of other strategies for two-stage stochastic programming: i) the DEP problem in Section 4.1.1, and ii) the two-stage stochastic programming based on GA in Section 4.1.2.

For all of the simulations, 20 different air traffic situations are randomly generated, where eight arriving aircraft through PMS are considered. First, in each air traffic situation, the type of aircraft is randomly assigned among heavy, large, and small. Second, two routes r_L and r_R, as shown in Fig. 2.1, are randomly assigned to each aircraft. Third, the estimated time of arrival at the initial point of PMS, i.e., T^{ETA}_{f,r,p_I}, is randomly given. The lower and upper bounds of T^{ETA}_{f,r,p_I} are set to SEP_{f,f',p_I} and $2 \cdot SEP_{f,f',p_I}$, respectively. Note that the uncertainty of flight time during CDA can be modelled as a normal distribution through historical data analysis performed in Section 3.5. In particular, in the case of PMS operated in Jeju International Airport, the mean μ^{CDA} and standard deviation σ^{CDA} are 274.60 seconds and 27.75 seconds, respectively. All computations are performed by using MATLAB [81] and a desktop PC with an Intel Core (3.60 GHz) processor.

4.4.1 Numerical Analysis on the Number of Scenarios

The aim of the numerical simulation in this section is to determine the appropriate number of scenarios in the second stage of the proposed two-stage stochastic programming based on PSO. Despite recent research efforts [22, 23], the determination of the number of scenarios in the second stage of the two-stage stochastic programming based on evolutionary algorithms remains empirical and is also problem dependent. A large number of scenarios will increase the probability of finding a better solution which can fully reflect the characteristics of uncertainty; however, it requires higher computational loads in the second stage. For this reason, the appropriate number of scenarios, which can fully reflect the uncertainty with a low computational load, is desirable. In this study, to find the best tradeoff between the accuracy (the realization of uncertainty) and the efficiency (the computational load), following numerical simulations are performed.

First, 20 different air traffic situations are considered. For each air traffic situation, 150 scenarios are generated by considering the uncertainty of flight time during CDA. Note that one scenario is sampled from the normal distribution function, i.e., $N(\mu^{CDA}, \sigma^{CDA^2})$. The proposed two-stage stochastic programming based on PSO is evaluated with various numbers of scenarios in the second stage, i.e., 25, 50, $\cdots$, 125 scenarios. Finally, total flight times and computational times are compared. MATLAB optimization toolbox [81] is used to implement PSO. As summarized in Table 4.3, the acceleration constants c_1 and c_2 are set to default values, and the inertia weight W is gradually decreased within the range of [0.1 1.1].

Figures 4.3 and 4.4 show the total flight time with respect to the elapsed CPU time for one sample air traffic situation among the 20 different air traffic

Table 4.3 Parameters used in the first stage of the proposed two-stage stochastic programming based on PSO

	Parameters	Value
First stage	Number of variables (=nvars)	$N_F \cdot (N_P - 1)$
	Number of particles	$3 \cdot$ nvars
	Maximum time (hour)	6
	W in Eq. (4.22)	[0.1 1.1] (default)
	c_1 in Eq. (4.22)	1.49 (default)
	c_2 in Eq. (4.22)	1.49 (default)

situations, where the small circle, triangle, and square represent each iteration. As shown in Figs. 4.3 and 4.4, CPU time elapsed in each iteration increases as the number of scenarios considered in the second stage increases. Note that the obtained best value in the total flight time depends on the number of scenarios. Figure 4.4 shows the difference between the total flight times obtained by considering 100, 125, and 150 scenarios. The best values in the total flight time obtained by considering 25, 50, and 75 scenarios slightly deviate from the best value in the total flight time obtained by considering 150 scenarios, as shown in Fig. 4.3.

Figures 4.5 and 4.6 show the overall results for the 20 different air traffic situations. Figure 4.5 shows the total flight times using Whisker plots, where the central mark is the median, the edges of the box are the 25th and 75th percentiles, the top and bottom horizontal lines are the minimum and maximum data points, and the outliers are plotted individually. The differences between the median of 150 scenarios and that of other numbers of scenarios are summarized in Table 4.4. If the difference is small, then the corresponding number of scenarios indicates that it fully reflects the uncertainty. However, as shown in Table 4.4, the results of 25, 50, and 75 scenarios strongly deviate from those of 150 scenarios. Figure 4.6 shows the elapsed CPU time using the Whisker plots. It is apparent from Fig. 4.6 that more computational time is required

as the number of scenarios increases. Therefore, it can be concluded that 100 scenarios are appropriate for the proposed two-stage stochastic programming based on PSO because the results for 100 scenarios are quite similar to those of 150 scenarios, as shown in Fig. 4.5 and Table 4.4, and the computational load of 100 scenarios is much less than that of 150 scenarios, as shown in Fig. 4.6.

Table 4.4 Median of various numbers of scenarios and differences in the median from 150 scenarios

Number of scenarios	25	50	75	100	125	150
Median (sec)	4,420	4,419	4,379	4,382	4,380	4,383
Difference in the median (sec)	37.42	36.73	3.73	1.14	2.38	-

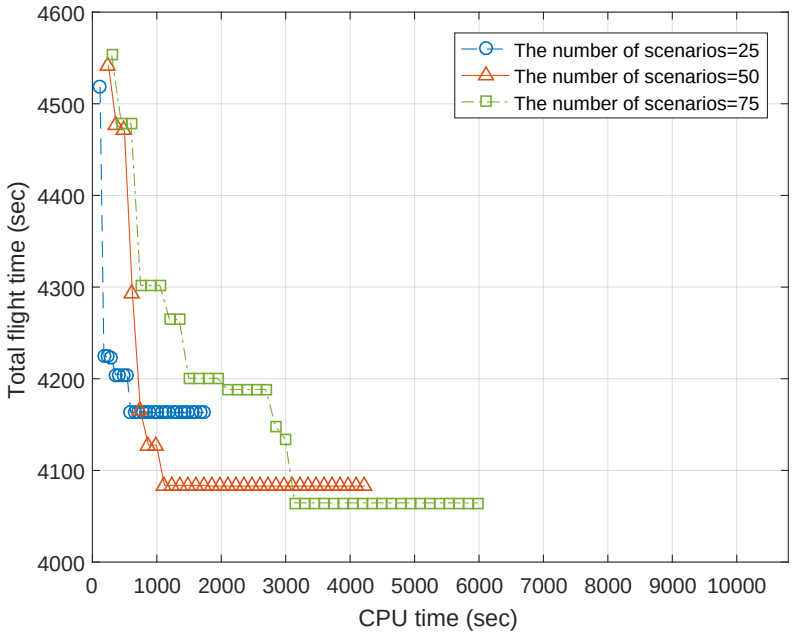


Figure 4.3 Total flight time with respect to CPU time for one situation when 25, 50, and 75 scenarios are considered

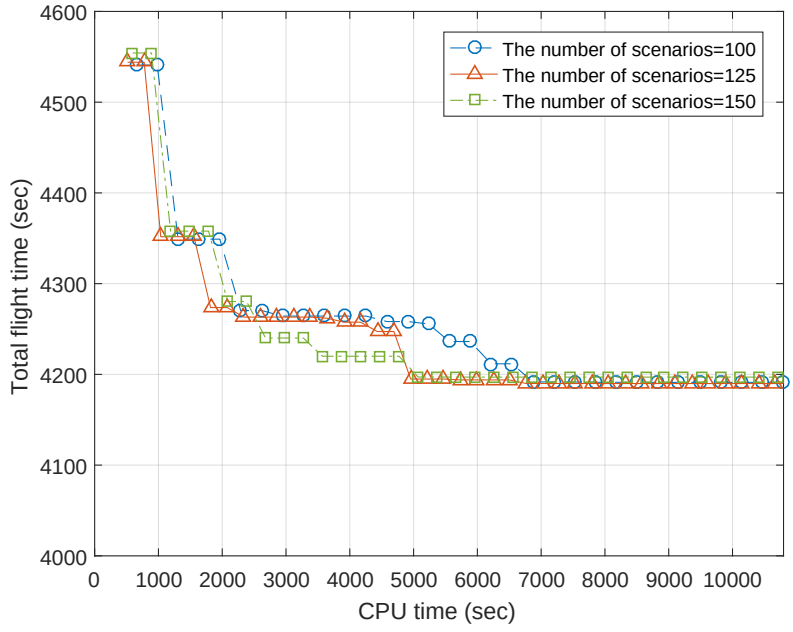


Figure 4.4 Total flight time with respect to CPU time for one situation, when 100, 125, and 150 scenarios are considered

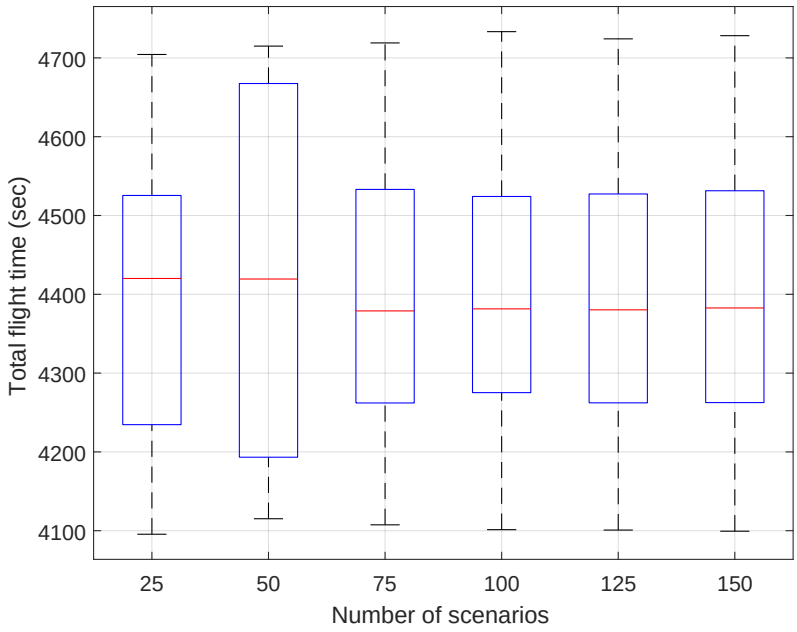


Figure 4.5 Total flight time for each number of scenarios

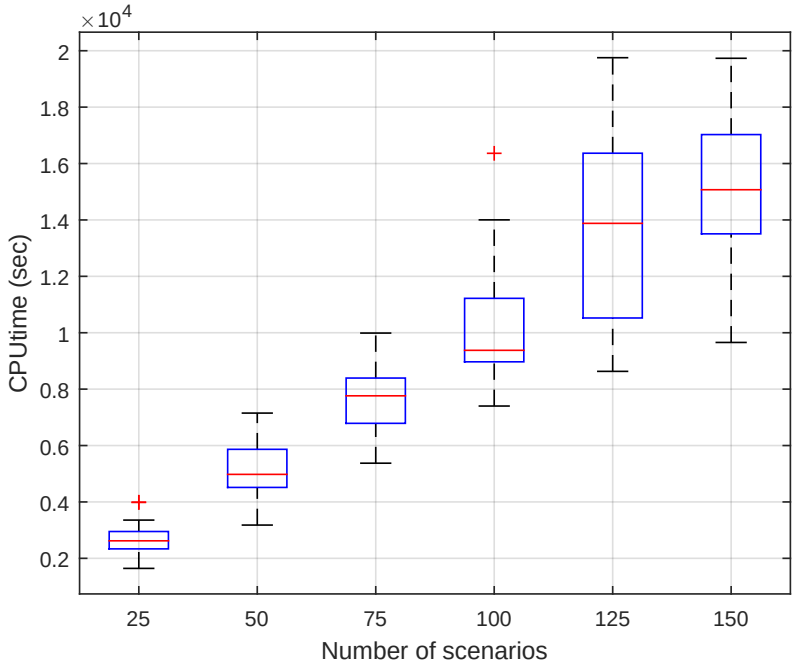


Figure 4.6 Elapsed CPU time for each number of scenarios

4.4.2 Comparison with Deterministic Programming

In this section, the performance of the proposed two-stage stochastic programming based on PSO is compared with that of the proposed deterministic programming in Chapter 3. The 20 different air traffic situations are considered for the simulation. Based on the simulation results in Section 4.4.1, the number of scenarios required in the second stage of the proposed two-stage stochastic programming based on PSO is set to 100.

Let us consider the heuristic adjustment in Section 3.3.1 to implement the proposed deterministic programming in dynamic environments. In the proposed deterministic programming, the uncertainty of flight time during CDA is considered to be the constant value, i.e., $\mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}$ in Eq. (3.7). Note that the constant value, especially the extra buffer size $\varepsilon\,\lambda\,\sigma^{CDA}$, is determined by the reliability level considered in the proposed deterministic programming. In this simulation, various reliability levels, i.e., 2.5%, 7.5%, $\cdots$, 22.5%, are considered to determine the extra buffer size. If the actual flight time during CDA of an aircraft is greater than the constant value $\mu^{CDA} + \varepsilon\,\lambda\,\sigma^{CDA}$, then its trailing aircraft's schedules are adjusted by using the heuristic adjustment in Section 3.3.1.

Likewise, the two-stage stochastic programming based on PSO can be implemented in dynamic environments as follows. First, for 100 different scenarios (i.e., 100 different realizations of uncertainty), the optimal aircraft sequence is determined by using the two-stage stochastic programming based on PSO. Let us consider the ordered list containing the sum of 100 different realizations of uncertainty. Then, by considering the 50th or 100th percentiles of the list, the optimal aircraft schedule is determined for the given optimal aircraft sequence. After the optimal aircraft sequence and schedule are determined, ac-

tual flight time during CDA is evaluated for a newly realized uncertainty. If the actual flight time during CDA of an aircraft is greater than the 50th or 100th percentiles of the ordered list, its trailing aircraft's schedules are adjusted to achieve the safe separation with the aircraft at the merge point. Note that this procedure is similar with the heuristic adjustment described in Section 3.3.1. Figure 4.7 shows the overall strategy to implement both the proposed deterministic programming and the proposed two-stage stochastic programming based on PSO in dynamic environments.

To compare the performance of the two-stage stochastic programming based on PSO and the deterministic programming, Monte Carlo simulations are performed 50 times for one air traffic situation. For each simulation, the actual flight time during CDA is newly sampled from the normal distribution, i.e., $N(\mu^{CDA}, \sigma^{CDA^2})$. The simulation results are shown in Fig. 4.8, where each data point is an average of 20 different air traffic situations. The left y-axis corresponds to the total flight time of eight arriving aircraft through PMS. The dark-colored bar against the left y-axis shows the total flight time before conducting the heuristic adjustment, and the light-colored bar against the left y-axis represents the increase in the total flight time caused by the heuristic adjustment. The line graph against the right y-axis corresponds to the number of aircraft on which a substantial uncertainty during CDA is imposed.

Let us now examine the results given in Fig. 4.8. First, from the results obtained by the deterministic programming with various reliability levels (from 2.5% to 22.5%), it can be stated that the total flight time increases but fewer heuristic adjustments are required as the reliability level decreases, i.e., as the extra buffer size increases. The additional flight time caused by the heuristic adjustment decreases as the reliability level decreases. The reason is that more time separation is imposed to compensate for the effects of the actual

uncertainty when the extra buffer size is small. Second, the comparison results between the two-stage stochastic programming based on PSO and the deterministic programming indicate that even though the uncertainty of flight time during CDA is modelled as a probabilistic distribution function in the two-stage stochastic programming based on PSO, the results do not deviate much from those of the deterministic programming. Before performing heuristic adjustment, the flight time determined by the two-stage stochastic programming based on PSO with the 50th percentile uncertainty becomes 4,821 seconds, which is less than that of deterministic programming with a 22.5% reliability level, i.e., 4,865 seconds. However, when the 100th percentile uncertainty is considered, the flight time determined by the two-stage stochastic programming based on PSO increases and becomes 5,050 seconds, but it is less than that of deterministic programming with a 2.5% reliability level, i.e., 5,136 seconds. Third, in case of the two-stage stochastic programming based on PSO with the 50th percentile uncertainty, the additional flight time caused by heuristic adjustment is 271 seconds which is greater than that of deterministic programming. The reason is that, when the 50th percentile uncertainty is considered, the number of substantial uncertainties is twice greater than that of deterministic programming as shown in the line graph. On the other hand, the smallest additional flight time, 147 seconds, can be determined by the two-stage stochastic programming based on PSO when the 100th percentile uncertainty is considered. It is considered that, although the considered worst uncertainty might be greater or less than the actual uncertainty, it does not deviate much from the actual uncertainty because it is sampled from the probability distribution.

In summary, the comparison results between the proposed deterministic programming and the proposed two-stage stochastic programming based on PSO depend on which uncertainty is considered among 100 different realizations

of uncertainty. However, when the 50th percentile uncertainty is considered, the total flight time, 5,092 seconds, is less than that of deterministic programming with a 17.5% reliability level, i.e., 5,099 seconds. In addition, when the 100th percentile uncertainty is considered, the total flight time, 5,197 seconds, is less than that of deterministic programming with a 2.5% reliability level, i.e., 5,315 seconds. Therefore, it can be concluded that the robust solutions of the proposed two-stage stochastic programming based on PSO are less conservative that those of the proposed deterministic programming with a 2.5% reliability level.

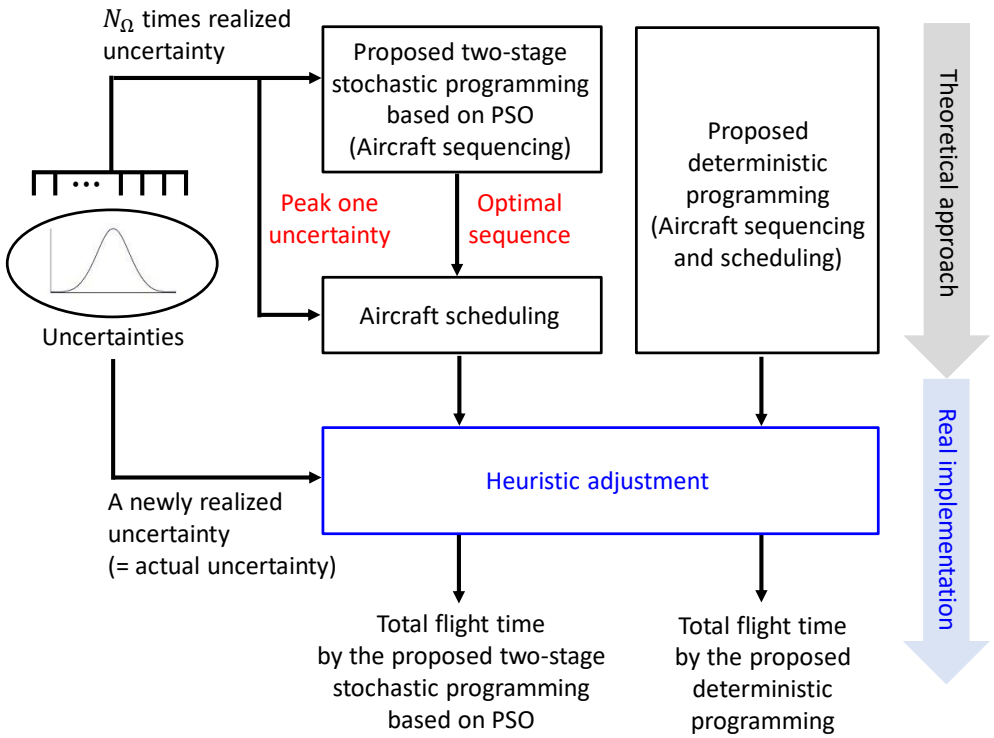


Figure 4.7 Strategy to implement both the proposed deterministic programming and the proposed two-stage stochastic programming based on PSO in dynamic environments

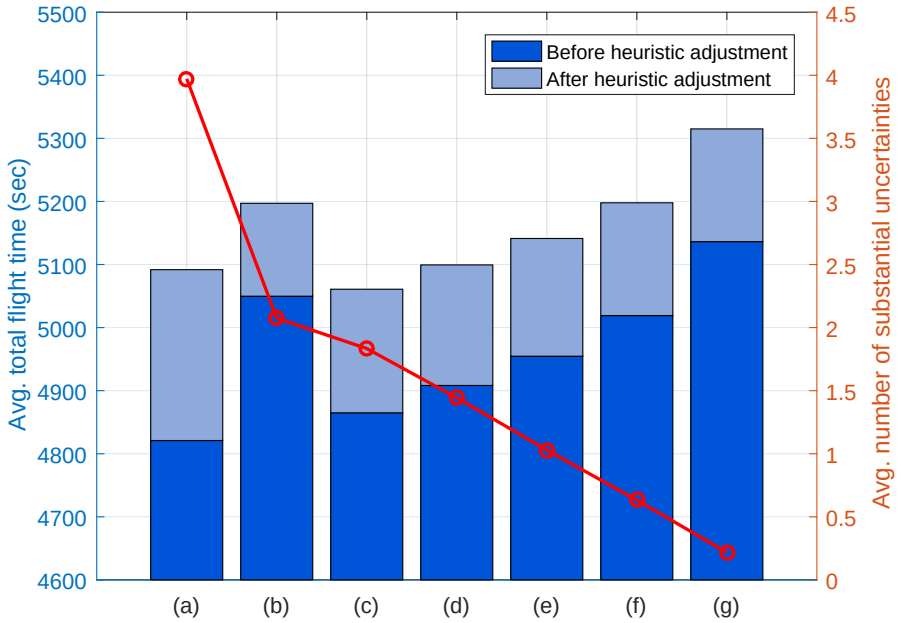


Figure 4.8 Average total flight time (left y-axis) and number of substantial uncertainties (right y-axis) comparisons between the proposed two-stage stochastic programming based on PSO with the (a) 50th and (b) 100th percentile uncertainties and the proposed deterministic programming with (c) 22.5%, (d) 17.5%, (e) 12.5%, (f) 7.5%, and (g) 2.5% reliability levels

4.4.3 Comparison with Other Stochastic Programming

In this section, the performance of the proposed two-stage stochastic programming based on PSO is compared with that of other strategies to solve two-stage stochastic programming: the DEP problem solved with CPLEX [66, 67] in Section 4.1.1 and the two-stage stochastic programming based on GA [22, 23] in Section 4.1.2. Note that MATLAB optimization toolbox [81] is used to implement PSO and GA. As summarized in Table 4.5, the crossover rate, mutation rate, and elitism rate, which are related to the convergence performance of GA, are set to default values. For the simulation, the previously generated 20 different air traffic situations are considered, and the number of scenarios required in the second stage is set to 100.

The comparison results between PSO and GA can be categorized into four cases; 1) PSO wins, 2) GA wins, 3) PSO wins but finds a sub-optimum, and 4) GA wins but finds a sub-optimum. The sample simulation results for each case are shown in Figs. 4.9–4.12, where the small circle in the solid line and the small triangle in the dotted line correspond to each iteration of PSO and each generation of GA, respectively, and the dashed line represents the exact optimum determined by CPLEX.

First, PSO converges to the exact optimum more quickly than GA for 9 of the 20 situations (45%). Figure 4.9 shows the total flight time versus the elapsed CPU time for this case. On the other hand, GA finds the exact optimum in a shorter time than PSO for 6 of the 20 situations (30%) as shown in Fig. 4.10. There are also some cases in which both PSO and GA fail to find the exact optimum within 3 hours of CPU time. In these cases, PSO finds a better sub-optimum than GA for 2 of the 20 situations (10%). However, the opposite results are obtained for 3 of the 20 situations (15%). In Table 4.6, the average deviation

between the exact optimum determined by CPLEX and the best values achieved by PSO and GA for the 20 situations are summarized according to the maximum CPU time. As shown in Table 4.6, for all cases, PSO finds the best value which is less deviated from the optimum determined by CPLEX than GA. In particular, within 0.5 hour of CPU time, PSO makes a significant difference compared to GA in the convergence performance. Although the difference is reduced when the CPU time limit is set to one hour, it becomes more substantial as the CPU time limit increases.

For all of the cases, PSO and GA find better solutions than CPLEX within 3 hours of CPU time. The reason is that, 56,000 variables are required in the DEP problem to formulate the aircraft arrival sequencing and scheduling problem in PMS by considering 100 scenarios, and therefore the elapsed CPU time is greater than 24 hours. It can be concluded that PSO and GA are more preferable than CPLEX when optimal or sub-optimal solutions are to be obtained within a short time. Furthermore, if it is required to determine sub-optimal solutions within a very short time (less than 0.5 hour of CPU time), then PSO is a better choice than other strategies for two-stage stochastic programming.

Table 4.5 Parameters used in the first stage of the two-stage stochastic programming based on GA

	Parameter	Value
First stage	Number of chromosome	$3 \cdot \text{nvars}$
	Maximum time (hour)	6
	Crossover rate	80% (default)
	Mutation rate	20% (default)
	Elitism rate	5% (default)

Table 4.6 Comparison results PSO vs. GA

Maximum CPU time (hour)	0.5	1	1.5	2	2.5	3
Average deviation of PSO (sec)	129.32	59.03	37.28	25.04	24.57	22.61
Average deviation of GA (sec)	154.13	60.79	38.28	37.15	37.15	37.15

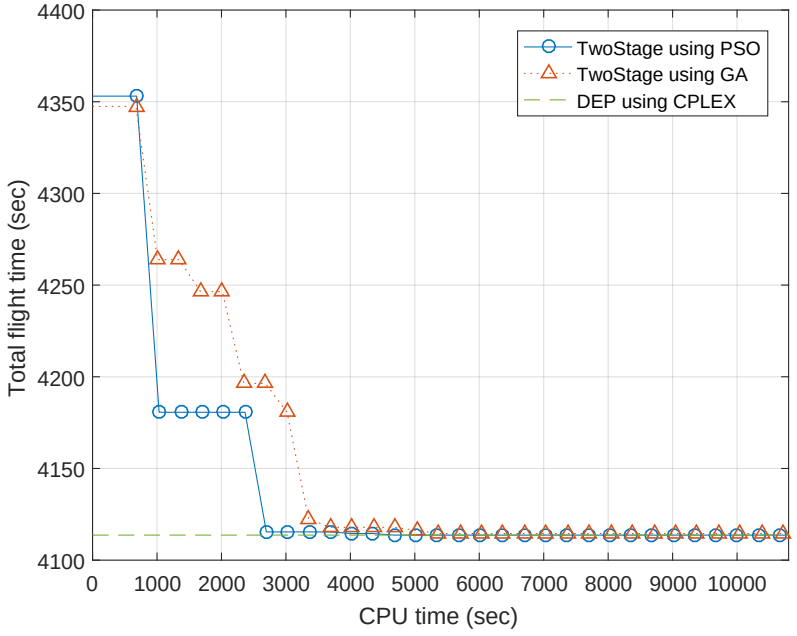


Figure 4.9 Total flight time with respect to CPU time for case 1

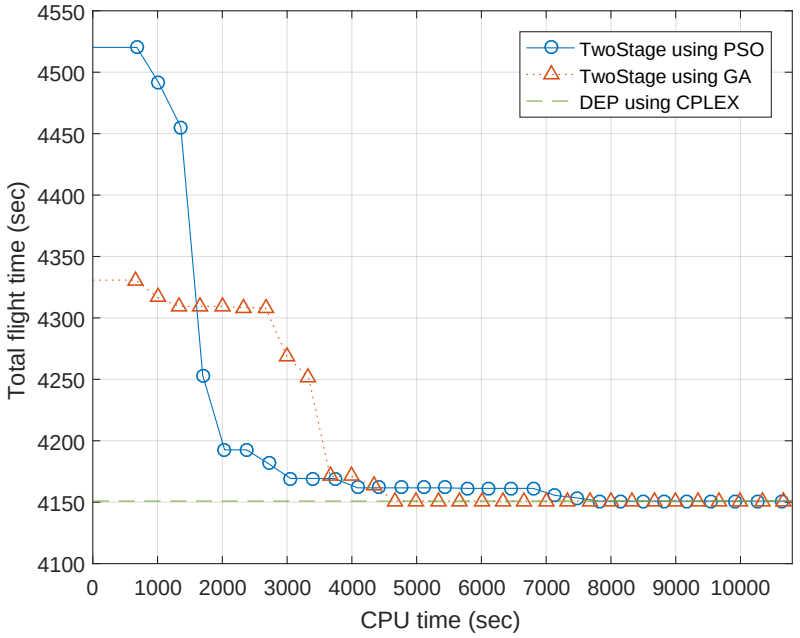


Figure 4.10 Total flight time with respect to CPU time for case 2

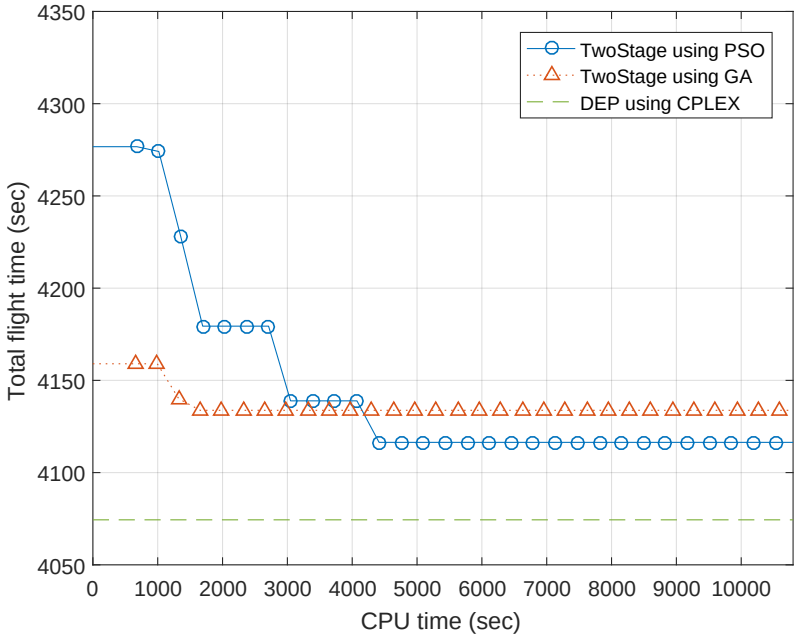


Figure 4.11 Total flight time with respect to CPU time for case 3

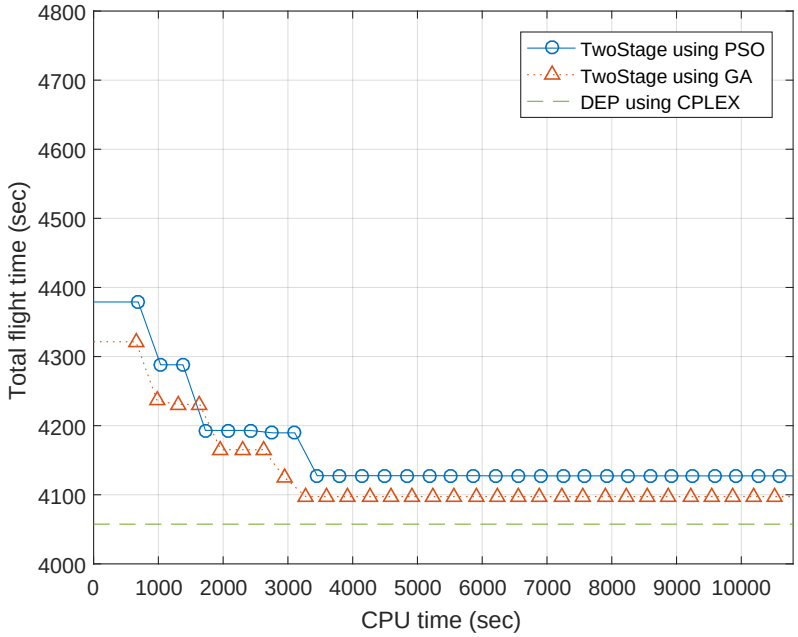


Figure 4.12 Total flight time with respect to CPU time for case 4

Chapter 5

Conclusions

5.1 Summary

In this dissertation, robust optimization methods for aircraft sequencing and scheduling in the Point Merge System (PMS), which can support the decision making of human air traffic controllers in real operation, were developed.

First, this study attempted to determine exact optimal aircraft sequence and schedule for each aircraft arriving through PMS based on Mixed Integer Linear Programming (MILP). In the proposed MILP formulation, the objective function was designed to minimize the total flight time that is required for all aircraft to reach the merge point, and four constraint equations were derived by considering the arrival procedure through PMS. Although the typical aircraft sequencing and scheduling problem requires a large amount of computational time when the numbers of aircraft, points, and routes increase, the proposed MILP formulation achieved a reduction in the number of variables by considering the typical configuration and characteristics of PMS. Therefore, the proposed MILP formulation could be efficiently implemented without significant computational effort.

Second, to consider the uncertainty of flight time during Continuous Descent Approach (CDA), robust optimization based on deterministic programming was proposed. The extra buffer size was analytically derived based on the determin-

istic robust counterpart problem for given feasibility tolerance and reliability level. Unlike the typical deterministic programming, of which robust solutions are only available in static environments, alternative solutions were determined to compensate for unexpected situations under dynamic environments by using the proposed heuristic adjustment. In addition, the operational time of the proposed deterministic programming was extended with a low computational load by using the sliding time window. By performing the historical data analysis, only 62% of aircraft arriving through the PMS used in Jeju International Airport were precisely maneuvered during a nominal day; therefore, the historical data analysis results suggested that decision support tools for human air traffic controllers in PMS are required to increase the control capacity of the current air traffic control system. The performance of the proposed deterministic programming was validated through historical flight data during the peak hour of Jeju International Airport. Among the various reliability levels, a 17.5% reliability level was considered as the most appropriate reliability level because the resulting total flight time was reduced to the minimum and the percent decrease in the number of heuristic adjustments from the maximum was 40.0% (=100×(5-3)/5). When a 17.5% reliability level was considered, 42 calculations were performed with a sliding time window; as a result, the maximum CPU time required for each calculation was maintained to less than 0.62 seconds. Therefore, it can be stated from the simulation results that the proposed deterministic programming can be efficiently implemented over a long time period (approximately one hour) with a low computational load.

Finally, two-stage stochastic programming based on Particle Swarm Optimization (PSO) was proposed for the aircraft sequencing and scheduling problem in PMS to compensate for the uncertainty that arises in the CDA trajectory. Unlike the previous studies on two-stage stochastic programming, which have

focused on approximation and decomposition methods, PSO was utilized for the randomized search to make the first stage decision, i.e., the optimal aircraft sequence, under incomplete information about uncertain parameters. In addition, the second stage decision, i.e., the optimal aircraft schedule, was made by solving MILP after the realization of uncertain parameters. Because of the continuous nature of PSO, the random key representation was utilized to convert the continuous aircraft position values to the discrete aircraft sequence. Numerical analysis was performed to determine the appropriate number of scenarios required in the second stage of the proposed two-stage stochastic programming process. Consequently, 100 scenarios were appropriate because the difference in the median from 150 scenarios was less than 2 seconds and the percent decrease in CPU time from 150 scenarios was 37.8% (=100×(15,071-9,376)/15,071). Additionally, the performance of the proposed two-stage stochastic programming based on PSO was evaluated and compared with that of the proposed deterministic programming and other stochastic approaches. In comparison with the proposed deterministic programming, the percent decrease in the total flight time from the proposed deterministic programming with a 2.5% reliability level to the proposed two-stage stochastic programming based on PSO was 4.2% (=100×(5,315-5,092)/5,315) and 2.2% (=100×(5,315-5,197)/5,315), when the 50th and 100th percentile uncertainties are considered, respectively. In comparison with other stochastic approaches, for 55% of the 20 situations, the proposed two-stage stochastic programming based on PSO could find better solutions than the previous stochastic approach with Genetic Algorithm (GA). In particular, when the maximum CPU time is less than 0.5 hour, the percent decrease in the average difference between the exact optimum and the suboptimum from GA to PSO was 16.1% (=100×(154.1-129.3)/154.1). The results showed that the proposed two-stage stochastic programming based on PSO is

less conservative than the proposed deterministic programming and preferable than other stochastic approaches when a short computing time is allowed.

5.2 Future Research Directions

The work presented in this dissertation can be expanded in the following directions in the future.

5.2.1 Applications of Multi-Objective Optimization

While this study focuses on a single-objective optimization problem to determine the optimal solutions from the perspective of air traffic control, i.e., total flight time minimization, a multi-criteria decision-making problem can further be considered for the aircraft sequencing and scheduling problem because of the following reasons: multiple stakeholders, including airport, airlines, and human air traffic controllers, are involved in the aircraft sequencing and scheduling problem [50, 79, 86]; from the perspective of airport operations, minimizing the makespan (i.e., maximizing the throughput) is highly desirable [79]; airlines are interested in the economic benefit, such as fuel consumption and aircraft delay [79, 86]; human air traffic controllers aim at minimizing the number of conflicts because their highest priority is the safety [50, 86]. Therefore, each criterion should be carefully modeled to reflect various perspectives of different stakeholders.

Instead of determining a single optimal solution, the new aim of the multi-objective optimization problem is to find the set of Pareto optimal solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the others [87]. To determine the Pareto optimal solutions that realize a good tradeoff among the objective functions, the following optimization techniques have been developed: weighted sum method [88], ε-constraint method [89], and multi-objective evolutionary algorithm [90]. Therefore, in an extended study, it would be interesting to adopt the existing optimization techniques for the multi-objective optimization problem or to develop

a new method for accurate and efficient determination of the Pareto optimal solutions.

5.2.2 Extensions of Airport Surface Traffic Optimization

In this study, the robust optimization for aircraft arrival sequencing and scheduling was independently developed and did not consider the interaction with departures and airport surface operations. However, further investigation into the integrated arrival, departure, and surface operations is strongly recommended to improve the efficiency because the schedule of arrivals and departures might be affected by ground conditions [21, 91].

As a solution methodology, the final scheduling point of this study, i.e., the merge point of PMS, could be shifted to the gate on the airport surface by including runways as well as spots along taxiways on the airport surface as the intermediate scheduling points [92, 93]. In addition, several important operational constraints regarding airport surface operations should also be addressed [94]. Therefore, the integrated optimization model might require a long computation time to determine the exact optimal solutions because the number of aircraft, points, and routes at the same time period inevitably increases. To be computationally tractable, further studies are required to develop the integrated optimization model as a unified single system [95] or sequentially connected optimization systems [79]; in addition, further studies could also be conducted to implement different optimization techniques.

5.2.3 Consideration of Various Uncertainties

A limitation of this study is that one uncertainty, which arises in the CDA trajectory, was considered for robust optimization. However, as mentioned previously, uncertainty arises from various sources in the complex terminal area; for example, the estimated time of arrival might be inaccurate, and the speed

of aircraft might not be maintained constantly in the sequencing leg. For this reason, further investigation of robust optimization considering various uncertainties is strongly recommended for aircraft sequencing and scheduling.

First, to consider various uncertainties in robust optimization based on deterministic programming, the deterministic robust counterpart problem should be newly derived when uncertainty arises from the coefficients and the right-hand-side parameters of the inequality constraints as well as the coefficients of the objective function [15]. In addition, for robust optimization based on stochastic programming, the decision variables should be carefully divided into the first and second stage decisions as various uncertainties are considered. Additionally, to fully reflect all characteristics of various uncertainties, a large number of scenarios should be considered in robust optimization.

Furthermore, future studies can be undertaken to perform sensitivity analysis for various uncertainties. Various methodologies for sensitivity analysis have been investigated in the fields of computer and chemical engineering [96–98]. For example, in [96], the change in the objective function value with respect to the change of the protection level (i.e., the extra buffer size) was estimated because the degree of protection controls the tradeoff between the provability of violation and the effect to the objective function of the nominal optimization problem. In addition, in [97], analytical expressions for the sensitivities were derived, and then the sensitivities of the objective function with respect to various uncertainties were calculated to determine the most sensitive uncertainty. Because a few works have been reported in the fields of ATM, further studies regarding sensitivity analysis with various uncertainties would be worthwhile for the aircraft sequencing and scheduling problem.

Bibliography

[1] Bureau of Transportation Statistics, "Airline On-Time Statistics and Delay Causes," 2015. [Online] Available from: http://www.transtats.bts.gov/.

[2] ICAO, "Procedures for Air Navigation Services - Aircraft Operations (ICAO Doc 8168)," 2006. [Online] Available from: http://code7700.com/pdfs/icao-doc-8168-vol2.pdf.

[3] EUROCONTROL, "Arrival Manager," 2010. [Online] Available from: https://www.eurocontrol.int/sites/default/files/article/content/documents/nm/fasti-aman-guidelines-2010.pdf.

[4] EUROCONTROL, "PHARE Advanced Tools Departure Manager," 1999. [Online] Available from: https://www.eurocontrol.int/phare/gallery/content/public/documents/98-70-18-v5-dm.pdf.

[5] Favennec, B., Rognin, L., Trzmiel, A., Vergne, F., and Zeghal, K., "Point Merge in Extended Terminal Area," EEC Technical/Scientific Report No. 2010-011, Oct. 2010. [Online] Available from: http://www.eurocontrol.int/point-merge-extended-terminal-area.

[6] Rathinam, S., Montoya, J., and Jung, Y., "An Optimization Model for Reducing Aircraft Taxi Times at the Dallas Fort Worth International Airport," *26th International Congress of the Aeronautical Sciences (ICAS)*, Anchorage, AK, Sep. 2008.

[7] 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.

[8] Beasley, J. E., 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.

[9] Montoya, J., Rathinam, S., and Wood, Z., "Multiobjective Departure Runway Scheduling using Dynamic Programming," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 1, 2014, pp. 399–413.

[10] Heidt, A., Helmke, H., Liers, F., and Martin, A., "Robust Runway Scheduling Using a Time-Indexed Model," *Fourth SESAR Innovation Days*, Madrid, Spain, Nov. 2014.

[11] Heidt, A., Helmke, H., Kapolke, M., Liers, F., and Martin, A., "Robust Runway Scheduling Under Uncertain Conditions," *Journal of Air Transport Management*, Vol. 56, No. 1, 2016, pp. 28–37.

[12] Choi, S., Mulfinger, D. G., Robinson III, 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. 1–14.

[13] Xue, M., Zelinski, S., and Mulfinger, D., "Uncertainty Analysis of Integrated Departures and Arrivals: A Los Angeles Case Study," *Aviation Technology, Integration, and Operations Conference*, Los Angeles, CA, Aug. 2013.

[14] Ben-Tal, A., and Nemirovski, A., "Robust Solutions of Linear Programming Problems Contaminated with Uncertain Data," *Mathematical Programming*, Vol. 88, No. 3, 2000, pp. 411–424.

[15] Lin, X., Janak, S. L., and Floudas, C. A., "A New Robust Optimization Approach for Scheduling Under Uncertainty: I. Bounded Uncertainty," *Computers and Chemical Engineering*, Vol. 28, No. 6, 2004, pp. 1069–1085.

[16] Janak, S. L., Lin, X., and Floudas, C. A., "A New Robust Optimization Approach for Scheduling under Uncertainty: II. Uncertainty with Known Probability Distribution," *Computers and Chemical Engineering*, Vol. 31, No. 3, 2007, pp. 171–195.

[17] Brinton, C., and Atkins, S., "A Probabilistic Modeling Foundation for Airport Surface Decision Support Tools," *Integrated Communications, Navigation and Surveillance Conference*, Arlington, VA, May 2009.

[18] 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.

[19] Bosson, C., Xue, M., and Zelinski, S., "Optimizing Integrated Terminal Airspace Operations under Uncertainty," *Digital Avionics Systems Conference*, Colorado Springs, CO, Oct. 2014.

[20] Bosson, C., Xue, M., and Zelinski, S., "Optimizing Integrated Arrival, Departure and Surface Operations under Uncertainty," *USA/Europe ATM R&D Seminar*, Lisbon, Portugal, 2015.

[21] Bosson, C., Optimizing Integrated Airport Surface and Terminal Airspace Operations under Uncertainty, Ph.D. Dissertation, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, Dec. 2015.

[22] Till, J., Sand, G., Urselmann, M., and Engell, S., "A Hybrid Evolutionary Algorithm for Solving Two-Stage Stochastic Integer Programs in Chemical Batch Scheduling," *Computers and Chemical Engineering*, Vol. 31, No. 5, 2007, pp. 630–647.

[23] Zhou, Z., Zhang, J., Liu, P., Li, Z., Georgiadis, M. C., and Pistikopoulos, E. N., "A Two-Stage Stochastic Programming Model for the Optimal Design of Distributed Energy Systems," *Applied Energy*, Vol. 103, 2013, pp. 135–144.

[24] Idris, H., Delcaire, B., Anagnostakis, I., Hall, W., Pujet, N., Feron, E., Hansman, R. J., Clarke, J. P., and Odoni, A., "Identification of Flow Constraint and Control Points in Departure Operations at Airport Systems," *AIAA Guidance, Navigation, and Control Conference*, Boston, MA, Aug. 1998.

[25] Balakrishnan, H., and Chandran, B. G., "Algorithms for Scheduling Runway Operations under Constrained Position Shifting," *Operations Research*, Vol. 58, No. 6, 2010, pp. 1650–1665.

[26] Gupta, G., Malik, W., and Jung, Y., "A Mixed Integer Linear Program for Airport Departure Scheduling," *AIAA Aviation Technology, Integration, and Operations Conference*, Hilton Head, SC, Sep. 2009.

[27] Gupta, G., Malik, W., and Jung, Y., "Incorporating Active Runway Crossings in Airport Departure Scheduling," *AIAA Guidance, Navigation and Control Conference*, Toronto, Canada, Aug. 2010.

[28] 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.

[29] Bianco, L., Ricciardelli, S., Rinaldi, G., and Sassano, A., "Scheduling Tasks with Sequence-Dependent Processing Times," *Naval Research Logistics*, Vol. 35, No. 2, 1988. pp. 177–184.

[30] Bianco, L., Rinaldi, G., and Sassano, A., "A Combinatorial Optimization Approach to Aircraft Sequencing Problem," *Flow Control of Congested Networks*, Vol. 38, Springer, Berlin, Germany, 1987, pp. 323–339.

[31] Pujet, N., Delcaire, B., and Feron, E., "Input-Output Modeling and Control of the Departure Process of Busy Airports," *Air Traffic Control Quarterly*, Vol. 8, No. 1, 2000, pp. 1–32.

[32] Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D., "Scheduling Aircraft Landings–the Static Case," *Transportation Science*, Vol. 34, No. 2, 2000, pp. 180–197.

[33] Beasley, J. E., Sonander, J., and Havelock, P., "Scheduling Aircraft Landings at London Heathrow Using a Population Heuristic," *Journal of the Operational Research Society*, Vol .52, No. 1, 2001, pp. 483-493.

[34] Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D., "Displacement Problem and Dynamically Scheduling Aircraft Landings," *Journal of the Operational Research Society*, Vol. 55, No. 1, 2004, pp. 54–64.

[35] Dear, R. G., The Dynamic Scheduling of Aircraft in the Near Terminal Area, Ph.D. Dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, Sep. 1976.

[36] Dear, R. G., and Sherif, Y. S., "An Algorithm for Computer Assisted Sequencing and Scheduling of Terminal Area Operations," *Transportation Research Part A: General*, Vol. 25, No. 2, 1991, pp. 129–139.

[37] Hong, Y., Choi, B., Lee, S., Lee, K., and Kim, Y., "Optimal and Practical Aircraft Sequencing and Scheduling for Point Merge System," *20th IFAC World Congress*, Toulouse, France, July 2017.

[38] 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.

[39] 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.

[40] 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.

[41] Hu, X. B., and Di Paolo, E., "An Efficient Genetic Algorithm with Uniform Crossover for Air Traffic Control," *Computers and Operations Research*, Vol. 36, No. 1, 2009, pp. 245–259.

[42] Hu, X. B., and Di Paolo, E. A., "A Ripple-Spreading Genetic Algorithm for the Aircraft Sequencing Problem," *Evolutionary Computation*, Vol. 19, No. 1, 2011, pp. 77–106.

[43] Capozzi, B., Atkins, S., and Choi, S., "Towards Optimal Routing and Scheduling of Metroplex Operations," *AIAA Aviation Technology, Integration, and Operations Conference*, Hilton Head, SC, Sep. 2009.

[44] Capozzi, B., and Atkins, S., "A Hybrid Optimization Approach to Air Traffic Management for Metroplex Operations," *AIAA Aviation Technology, Integration, and Operations Conference*, Forth Worth, TX, Sep. 2010.

[45] Xue, M., and Zelinski, S., "Optimal Integration of Departures and Arrivals in Terminal Airspace," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 1, 2013, pp. 207–213.

[46] Boursier, L., Favennec, B., Hoffman, E., Trzmiel, A., Vergne, F., and Zeghal, K., "Merging Arrival Flows without Heading Instructions," *7th USA/Europe Air Traffic Management R&D Seminar*, Barcelona, Spain, July 2007.

[47] Favnnec, 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," *AIAA Aviation Technology, Integration, and Operations Conference*, Hilton Head, SC, Sep. 2009.

[48] Ivanescu, D., Shaw, C., Tamvaclis, C., and Kettunen, T., "Models of Air Traffic Merging Techniques: Evaluating Performance of Point Merge," *AIAA Aviation Technology, Integration, and Operations Conference*, Hilton Head, SC, Sep. 2009.

[49] Sahin Meric, Ö., and Usanmaz, O., "A New Standard Instrument Arrival: the Point Merge System," *Aircraft Engineering and Aerospace Technology*, Vol. 85, No. 2, 2012, pp. 136–143.

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

[51] Liang, M., Delahaye, D., and Xu, X. H., "A Novel Approach to Automated Merge 4D Arrival Trajectories for Multi-Parallel Runways," *4th ENRI International Workshop on ATM/CNS*, Tokyo, Japan, Nov. 2015.

[52] Liang, M., Delahaye, D., Sbihi, M., and Ma, J., "Multi-Layer Point Merge System for Dynamically Controlling Arrivals on Parallel Runways," *IEEE/AIAA Digital Avionics Systems Conference*, Sacramento, CA, Sep. 2016.

[53] Liang, M., Delahaye, D., and Marechal, P., "Potential Operational Benefits of Multi-Layer Point Merge System on Dense TMA Operation Hybrid Arrival Trajectory Optimization Applied to Beijing Capital International Airport," *International Conference on Research in Air Transportation*, Philadelphia, PA, June 2016.

[54] Hong, Y., Lee, S., Lee, K., and Kim, Y., "Optimal Scheduling Algorithm for Air Traffic Point Merge System Using MILP," *4th CEAS Specialist Conference on Guidance, Navigation and Control*, Warsaw, Poland, Apr. 2017.

[55] Atkin, J. A., 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.

[56] Agogino, A., and Rios, J., "Robustness of Two Air Traffic Scheduling Approaches to Departure Uncertainty," *Digital Avionics Systems Conference*, Seattle, WA, Oct. 2011.

[57] Mueller, E. R., and Chatterji, G. B., "Analysis of Aircraft Arrival and Departure Delay Characteristics," *Aviation Technology, Integration, and Operations Conference*, Los Angeles, CA, Oct. 2002.

[58] Solveling, G., Stochastic Programming Methods for Scheduling of Airport Runway Operations under Uncertainty, Ph.D. Dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, Aug. 2012.

[59] Birge, J. R., and Louveaux, F., *Introduction to Stochastic Programming*, Springer, New York, NY, 1997.

[60] Benders, J. F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik*, Vol. 4, No. 1, 1962, pp. 238–252.

[61] Slyke, R. M. V., and Wets, R., "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming," *SIAM Journal on Applied Mathematics*, Vol. 17, No. 4, 1969, pp. 638–663.

[62] Shapiro, A., and Homem-de-Mello, T., "On the Rate of Convergence of Optimal Solutions of Monte Carlo Approximations of Stochastic Programs," *SIAM Journal on Optimization*, Vol. 11, No. 1, 2000, pp. 70–86.

[63] Kleywegt, A. J., Shapiro, A., and Homem-de-Mello, T., "The Sample Average Approximation Method for Stochastic Discrete Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 2, 2002, pp. 479–502.

[64] Shapiro, A., "Inference of Statistical Bounds for Multistage Stochastic Programming Problems," *Mathematical Methods of Operations Research*, Vol. 58, No. 1, 2003, pp. 57–68.

[65] Linderoth, J., Shapiro, A., and Wright, S., "The Empirical Behavior of Sampling Methods for Stochastic Programming," *Annals of Operations Research*, Vol. 142, No. 1, 2006, pp. 215–241.

[66] Hemmecke, R., and Schultz, R., "Decomposition Methods for Two-Stage Stochastic Integer Programs," *Online Optimization of Large Scale Systems*, Springer, Berlin, Germany, 2001, pp. 601–622.

[67] Escudero, L. F., Garin, M. A., Merino, M., and Perez, G., "An Exact Algorithm for Solving Large-Scale Two-Stage Stochastic Mixed-Integer Problems: Some Theoretical and Experimental Aspects," *European Journal of Operational Research*, Vol. 204, No. 1, 2010, pp. 105-116.

[68] Carøe, C. C., and Schultz, R., "Dual Decomposition in Stochastic Integer Programming," *Operations Research Letters*, Vol. 24, No. 1, 1999, pp. 37–45.

[69] Sand, G., and Engell, S., "Modeling and Solving Real-Time Scheduling Problems by Stochastic Integer Programming," *Computers and Chemical Engineering*, Vol. 28, No. 6, 2004, pp. 1087–1103.

[70] Dorigo, M., Birattari, M., and Stutzle, T., "Ant Colony Optimization," *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, 2006, pp. 28–39.

[71] Parsopoulos, K. E., and Vrahatis, M. N., "Recent Approaches to Global Optimization Problems through Particle Swarm Optimization," *Natural Computing*, Vol. 1, No. 2–3, 2002, pp. 235–306.

[72] Dong, Y., Tang, J., Xu, B., and Wang, D., "An Application of Swarm Optimization to Nonlinear Programming," *Computers and Mathematics with Applications*, Vol. 49, No. 11, 2005, pp. 1655–1668.

[73] Shi, Y., and Eberhart, R. C., "Empirical Study of Particle Swarm Optimization," *Congress on Evolutionary Computation*, Washington, DC, July 1999.

[74] Abido, M. A., "Optimal Power Flow Using Particle Swarm Optimization," *International Journal of Electrical Power and Energy Systems*, Vol. 24, No. 7, 2002, pp. 563–571.

[75] Bean, J. C., "Genetic Algorithms and Random Keys for Sequencing and Optimization," *ORSA Journal on Computing*, Vol. 6, No. 2, 1994, pp. 154–160.

[76] Tasgetiren, M. F., Liang, Y. C., Sevkli, M., and Gencyilmaz, G., "A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in the Permutation Flowshop Sequencing Problem," *European Journal of Operational Research*, Vol. 177, No. 3, 2007, pp. 1930–1947.

[77] Meric, O. S., and Turan, O., "Evaluation of Aircraft Descent Profile," *Energy Procedia*, Vol. 95, No. 1, 2016, pp. 308–313.

[78] Neuman, F., and Erzberger, H., "Analysis of Delay Reducing and Fuel Saving Sequencing and Spacing Algorithms for Arrival Spacing," NASA Technical Memorandum 103880, Oct. 1991.

[79] 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.

[80] de Castro Fortes, J. L., Pamplona, D. A., and Muller, C., "Analysis of Flight Trajectories in a Terminal Maneuvering Area: Sao Paulo-Rio de Janeiro Route as a Case Study," *International Journal of Science and Engineering Investigations*, Vol. 4, No. 42, 2015, pp. 80–85.

[81] MATLAB, ver. 9.0 (2016a), The MathWorks Inc., Natick, MA, 2016.

[82] IBM ILOG CPLEX V12.7: User's Manual for CPLEX, International Business Machines Corporation, Armonk, NY, 2016.

[83] Shi, Y., and Eberhart, R. C., "Parameter Selection in Particle Swarm Optimization," *International Conference on Evolutionary Programming*, Springer, Berlin, Germany, 1998, pp. 591–600.

[84] Maurice, C., and Kennedy, J., "The Particle Swarm Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transaction on Evolutionary Computation*, Vol. 6, No. 1, pp. 58–73.

[85] Laskari, E. C., Parsopoulos, K. E., and Vrahatis, M. N., "Particle Swarm Optimization for Integer Programming," *Proceedings of the Congress on Evolutionary Computation*, Vol. 2, No. 1, 2002, pp. 1582–1587.

[86] Yang, Y., Prandini, M., Cao, X., and Du, W., "A Multi-Criteria Decision-Making Scheme for Multi-Aircraft Conflict Resolution," *20th IFAC World Congress*, Toulouse, France, July 2017.

[87] Coello, C. A. C., Pulido, G. T., and Lechuga, M. S., "Handling Multiple Objectives with Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, 2004, pp. 256–279.

[88] Ghoseiri, K., Szidarovszky, F., and Asgharpour, M. J., "A Multi-Objective Train Scheduling Model and Solution," *Transportation Research Part B: Methodological*, Vol. 38, No. 10, 2004, pp. 927–952.

[89] Mavrotas, G., "Effective Implementation of the ε-constraint Method in Multi-Objective Mathematical Programming Problems," *Applied Mathematics and Computation*, Vol. 213, No. 2, 2009, pp. 455–465.

[90] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A. M. T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182–197.

[91] 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.

[92] 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.

[93] 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.

[94] Rathinam, S., Montoya, J., and Jung, Y. "An Optimization Model for Reducing Aircraft Taxi Times at the Dallas Fort Worth International Airport," *International Congress of the Aeronautical Sciences*, Anchorage, AK, Sep. 2008.

[95] Gilbo, E. P., "Optimizing Airport Capacity Utilization in Air Traffic Flow Management subject to Constraints at Arrival and Departure Fixes," *IEEE Transactions on Control Systems Technology*, Vol. 5, No. 5, 1997, pp. 490–503.

[96] Bertsimas, D., and Sim, M., "The Price of Robustness," *Operations Research*, Vol. 52, No. 1, 2004, pp. 35–53.

[97] Castillo, E., Conejo, A. J., Mínguez, R., and Castillo, C., "A Closed Formula for Local Sensitivity Analysis in Mathematical Programming," *Engineering Optimization*, Vol. 38, No. 1, 2006, pp. 93–112.

[98] Li, Z., and Ierapetritou, M., "Process Scheduling under Uncertainty: Review and Challenges," *Computers and Chemical Engineering*, Vol. 32, No. 4, 2008, pp. 715–727.

국문초록

항공기를 이용한 여행과 물류 수송을 위한 항공교통량이 급증함에 따라 항공교통 밀집을 해소하기 위한 연구의 필요성이 커지고 있다. 특히, 공항근처 공역은 도착하는 항공기와 출발하는 항공기가 집중되는 복잡한 공역으로, 많은 연구자들이 이 공역에서 항공교통관제사의 의사결정을 돕기 위한 항공기 시퀀싱 및 스케줄링 툴 개발의 필요성을 인지하고 있다. 그러나 공항근처 공역이 복잡하고 많은 불확실성 요소를 가지고 있음에도 불구하고 불확실성을 고려한 항공기 시퀀싱 및 스케줄링에 대한 연구는 미비한 실정이다.

본 연구에서는 항공기 시퀀싱 및 스케줄링 문제에서 불확실한 비행시간을 고려하기 위해 두 가지 강건 최적화 기법을 제안하였다. 첫 번째 기법은 결정론적 프로그래밍에 기반한 방법으로 2단계 계층적 구조를 갖는다. 상위 계층에서는 불확실성에 강건한 시퀀싱 및 스케줄링을 결정하기 위해 완충 값을 최적화 문제에 포함시키고, 완충 값의 크기를 수학적으로 유도하여 적용한다. 그러나 이러한 접근 방식은 실제 불확실성의 크기가 결정론적 프로그래밍에서 불확실성 파라미터를 대변하는 상수보다 작은 경우에만 유효하다. 따라서 하위 계층에서는 동적 환경에서도 강건한 항공기 시퀀싱 및 스케줄링을 계산하기 위해 상위 계층에서 결정된 강건한 시퀀싱 및 스케줄링 결과를 조정한다. 이를 위해 휴리스틱 조정 기법을 제안하였으며, 계산시간을 줄이기 위해 슬라이딩 시간 윈도우 개념을 도입하였다.

두 번째 기법은 확률론적 프로그래밍에 기반한 방법으로 입자 군집 최적화 알고리즘을 이용한 2단계 확률론적 프로그래밍 기법이다. 이 기법은 첫 번째 기법과 달리 불확실성을 확률분포함수 형태로 모델링하고, 이를 알고리즘상에서 고려한다. 따라서 결정론적 프로그래밍에 기반한 첫 번째 기법보다 덜 보수적인 시퀀싱 및 스케줄링을 결정할 수 있다. 확률론적 프로그래밍 기법의 첫 번째 단계에서는 강건한 항공기 시퀀싱을 결정하기 위해, 불확실성에 대한 정보가 없는 상태에서

입자 군집 최적화 알고리즘을 이용한 무작위 탐색을 수행한다. 두 번째 단계에서는 불확실성 정보를 고려하기 위해 첫 번째 단계에서 결정된 항공기 시퀀싱 결과를 보정하기 위한 항공기의 최적 스케줄링을 혼합정수선형계획법으로 결정한다. 본 연구에서 제안한 두 가지 강건 최적화 기법들의 성능은 실제 비행 데이터를 이용한 수치 시뮬레이션과 몬테카를로 시뮬레이션을 통해 검증하였다.

주요어: 항공기 시퀀싱 및 스케줄링, 강건 최적화, 혼합정수선형계획법, 결정론적 프로그래밍, 확률론적 프로그래밍

학번: 2012-30185