

## Original Article

## Point-Merge 절차를 이용한 도착 스케줄링 및 조언 정보 생성 알고리즘 개발

홍성권\*, 김소윤\*, 전대근\*, 은연주\*, 오은미\*

### A Development of Arrival Scheduling and Advisory Generation Algorithms based on Point-Merge Procedures<sup>†</sup>

Sungkweon Hong\*, Soyeun Kim\*, Daekeun Jeon\*, Yeonju Eun\*, and Eun-Mi Oh\*

#### ABSTRACT

This paper proposes arrival scheduling and advisory generation algorithms which can be used in the terminal airspace with Point-Merge procedures. The proposed scheduling algorithm consists of two steps. In the first step, the algorithm computes aircraft schedules at the entrance of the Point-Merge sequencing legs based on First-Come First-Served(FCFS) strategy. Then, in the second step, optimal sequence and schedules of all aircraft at the runway are computed using Multi-Objective Dynamic Programming(MODP) method. Finally, the advisories that have to be provided to the air traffic controllers are generated. To demonstrate the proposed algorithms, the simulation was conducted based on Jeju International Airport environments.

**Key Words** : Air Traffic Management(항공교통관리), Arrival Management(도착 관리), Scheduling(스케줄링), Multi-Objective Dynamic Programming(다목적합수 동적프로그래밍)

#### 1. 서 론

항공 교통량이 지속적으로 증가함에 따라 효율적인 항공교통관리 기법에 대한 관심이 나날이 증가하고 있다. 특히 여러 항공로로부터 특정 공항으로 도착하는 항공기들을 효율적으로 관리하기 위한 다양한 도착 관리(Arrival Management) 기법들이 연구되어왔다[1-3].

도착 관리기법은 동일한 공항으로 접근하는 항공기들을 대상으로 특정 스케줄링 지점에서의 분리 간격을 고려하여 효율적인 항공기 도착 순서 및 시각을 계산한 후 관련 조언 정보를 관제사에게 제공하는 것을 목적으로 한다. 하지만 일반적

인 도착 관리기법은 주로 접근관제공역에 진입하기 이전의 항공기를 대상으로 한다. 접근관제공역 내에서는 수평 및 수직적으로 인접한 다양한 절차가 동시에 운영되기 때문에, 일반적인 도착 관리 기법에서 제공되는 항공기의 순서 및 시각 정보를 관제사가 정확하게 준수하기 어렵다. 뿐만 아니라 제공된 조언 정보를 준수하기 위하여 관제사가 항공기의 경로를 임의로 변경할 경우, 항공기의 궤적 예측의 신뢰도가 낮아져 도착 관리의 효율성이 저하될 수 있다.

Liang et al.[4]은 이러한 문제점을 고려하여 접근관제공역 내에서 도착 관리를 제공하기 위한 방안으로서 Point-Merge 절차를 활용하는 방안과, 그에 따른 휴리스틱 기반의 스케줄링 알고리즘을 제안하였다. 이 연구에서는 Point-Merge 절차의 유연한 구조를 기반으로 항공기의 지연 시간을 처리하도록 함으로써 관제사의 조언 정보 처리를 수월하게 함과 동시에 항공기 궤적에 대한 예측성이 증대될 수 있을 것으로 기대하였다.

Received : 25. Aug. 2017. Revised : 20. Seo. 2017.

Accepted : 28. Sep. 2017

\* 한국항공우주연구원 차세대항행팀

연락처, E-mail : soyeun@kari.re.kr

대전광역시 유성구 과학로 169-84

본 연구에서는 Point-Merge 절차를 기반으로 한 도착 관리기법에서 사용될 수 있는 최적화 기반의 스케줄링 알고리즘을 제안하였다. 휴리스틱 방법을 기반으로 한 기존의 연구는 계산 속도가 빠른 장점이 있으나 최적화된 스케줄링 결과를 제공하지 못하는 한계점이 존재한다. 이러한 문제를 해결하기 위하여, 제안된 알고리즘은 기존 연구와 달리 Multi-Objective Dynamic Programming (MODP) 정식화를 통해 최적의 항공기 스케줄을 빠른 시간 안에 계산하는 것을 목적으로 한다. 본 논문의 2장에서는 일반적인 Point-Merge 절차에 대해 소개하였으며, 3장에서는 제안된 스케줄링 알고리즘에 대해 설명하였다. 4장에서는 제안된 알고리즘의 성능을 검증하기 위하여 수행한 시뮬레이션 결과를 제시하였으며, 5장에서는 결론과 향후 연구에 대해 논의하였다.

## 2. Point-Merge 절차

Fig. 1에는 일반적인 형태의 Point-Merge 절차가 나타나 있다. Point-Merge 절차는 일반적으로 하나의 direct-to point와 이를 중심으로 하는 두 개의 원호 형태인 sequencing leg로 이루어져 있다[5]. 각 leg는 수평 또는 수직적으로 분리되어 있으며, 여러 개의 절차가 동일한 고도로 합류하여 하나의 sequencing leg를 구성할 수 있다. 예를 들어, Fig. 1에 나타나 있는 절차의 경우 파란색 실선 및 빨간색 점선으로 표현된 두 개의 sequencing leg로 구성되어 있으며, 각 leg는 “A” 및 “B” merge fix에 두 개의 도착 절차가 동일한 고도로 합류하도록 되어 있다.

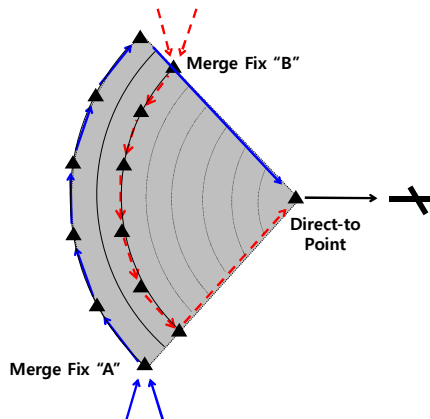


Fig. 1 General Point-Merge procedure

Point-Merge 절차를 통해 항공기를 관제하는 경

우, 선행 항공기가 존재하지 않는다면 sequencing leg를 비행하지 않고 바로 direct-to point로 비행하도록 할 수 있으며, 선행 항공기와의 분리가 필요한 경우 leg를 비행하다가 분리 간격이 충족되었을 때 direct-to point로 비행하도록 지시한다[5].

## 3. 스케줄링 알고리즘

제안된 스케줄링 알고리즘은 Fig. 2에 나타난 바와 같이 크게 두 단계로 나누어진다. 3.1절에 나열된 항공기 시간 정보가 주어진다고 가정하였을 때, 첫 번째 단계에서는 Point-Merge 절차의 각 merge fix로 수렴하는 항공기들을 대상으로 해당 픽스에서의 sequencing leg 진입 스케줄을 First-Come First-Served (FCFS) 순서 기반으로 결정한다. 두 번째 단계에서는 첫 번째 단계에서의 스케줄링 결과를 바탕으로 모든 항공기의 활주로 착륙 순서 및 스케줄을 MODP 알고리즘을 통해 결정한다. 두 번째 단계를 통해 각 leg 간 항공기의 착륙 순서를 FCFS 순서로부터 최적화된 순서로 변경할 수 있다. 또한 3.4절에서는 이러한 알고리즘 계산 결과를 관제사에게 제공하기 위한 조연 정보를 생성하는 방법을 소개하였다.

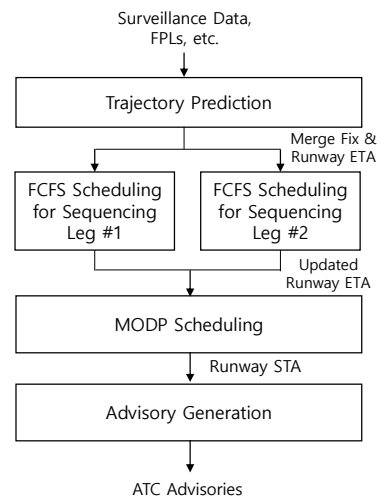


Fig. 2 Schematic view of proposed algorithms

### 3.1 항공기 예상 도착 시각 예측

본 연구에서는 스케줄링을 위한 항공기 예상 도착 시각(ETA: Estimated Time of Arrival)이 주어졌다고 가정하였다. 특정 항공기  $a$ 에 대하여

스케줄링에 사용되는 예상 도착 시각은 두 가지로, 첫 번째는 merge fix의 예상 도착 시각인  $ETA^F(a)$ 이며 두 번째는 sequencing leg 상의 각 fix에서 direct-to point로 비행한 경우의 활주로 도착 예상 시각 벡터인  $\overline{ETA}^R(a) = [ETA_1^R(a), ETA_2^R(a), \dots, ETA_P^R(a)]$ 이다. 여기서  $P$ 는 sequencing leg 상의 fix의 개수를 나타내며  $ETA_1^R(a)$  및  $ETA_P^R(a)$ 는 각각 항공기가 sequencing leg의 첫 번째 및 마지막 fix에서 direct-to point로 비행한 경우의 활주로 도착 예상 시각이다.

### 3.2 FCFS 스케줄링

FCFS 스케줄링은 merge fix들에 대하여 각각 독립적으로 수행된다.  $i$ 번째 sequencing leg의 merge fix에 수렴하는 항공기들이  $ETA^F$ 를 기준으로 정렬된 집합을  $A_i^F = [a_1^i, a_2^i, \dots, a_{N_i}^i]$ 라고 할 때, 해당 merge fix에서의 계획 도착 시각  $STA^F$ 는 첫 번째 항공기부터  $N_i$ 번째 항공기까지 다음과 같이 순차적으로 계산된다.

$$STA^F(a_j^i) = \begin{cases} ETA^F(a_j^i) & \text{if } j=1 \\ \max[ETA^F(a_j^i), STA^F(a_{j-1}^i) + sep(a_{j-1}^i, a_j^i)] & \text{otherwise} \end{cases} \quad (1)$$

여기서  $sep(a_{j-1}^i, a_j^i)$ 는 merge fix에서 충족되어야 하는 항공기  $a_{j-1}^i$ 와  $a_j^i$  간의 분리 간격이다.

식 (1)을 통해 모든 합류 지점에서의 계획 도착 시각이 계산되면, 다음과 같이  $\overline{ETA}^R$ 을 업데이트한다.

$$\overline{ETA}^R(a_j^i) \leftarrow \overline{ETA}^R(a_j^i) + FCFSDelay(a_j^i) \quad \forall i, j \quad (2)$$

여기서 업데이트된  $\overline{ETA}^R$ 은 FCFS 스케줄링을 통해 계산된  $STA^F$ 를 준수하였을 때 예상되는 활주로 도착 시각 벡터이며  $FCFSDelay$ 는  $STA^F$ 를 준수하기 위한 지연 시간으로 다음과 같이 계산된다.

$$FCFSDelay(a_j^i) = STA^F(a_j^i) - ETA^F(a_j^i) \quad (3)$$

### 3.3 MODP 스케줄링

FCFS 스케줄링을 통해 Point-Merge 절차의 각 merge fix에 대해 항공기 스케줄이 정해지면, 최종적으로 MODP 스케줄링 단계를 통해 활주로에 대한 도착 순서 및 스케줄이 결정된다. 이 단계에서는 동일한 sequencing leg를 통해 비행하는 항공기 간에 추월이 발생하지 않는 범위 내에서 최적의 항공기 스케줄을 찾는 것을 목적으로 한다.

MODP 스케줄링은 복수 개의 목적함수를 통해 최적의 스케줄을 계산하는 방법으로, 특정 항공기 시퀀스를 나타내는 상태변수와 각 목적함수에 대한 비용 함수가 정의되어야 한다[6-7]. Point-Merge 절차가 2개의 leg로 구성되어 있을 때, 특정 항공기 시퀀스  $s$ 에 대한 상태변수  $state(s)$ 는 다음과 같이 정의된다.

$$state(s) = (n, k_1, k_2) \in \Theta \quad (4)$$

여기서  $\Theta$ 는 모든 상태변수의 집합이고,  $n$ 은 항공기 시퀀스  $s$ 의 마지막 항공기가 몇 번째 leg를 통해 절차에 진입하는지를 나타내며,  $k_i$ 는  $s$ 에 포함된 항공기 중  $i$ 번째 sequencing leg를 통해 비행하게 될 항공기의 대수를 나타낸다. 또한 시퀀스  $s$ 가 항공기를 한 대도 포함하고 있지 않은 경우, 해당 시퀀스는 상태변수를  $(0, 0, 0)$ 으로 갖는다.

본 연구에서는 목적함수로서 마지막 항공기의 착륙 시각(makespan) 최소화 및 총 지연 시간(total delay) 최소화를 고려하였다.  $i$ 번째 sequencing leg로 비행하게 될 항공기의 대수가  $N_i$ 라고 할 때, 이 항공기들이 해당 leg의 첫 번째 fix에서 direct-to point로 비행을 할 경우에 대한 활주로 예상 도착 시각, 즉  $ETA_1^R$ 을 기준으로 정렬된 집합을  $A_i^R = [a_1^i, a_2^i, \dots, a_{N_i}^i]$ 라고 하자. 또한 특정 항공기 시퀀스  $s$ 와,  $s$ 에 한 대의 항공기가 추가된 시퀀스  $s'$ 가 있고, 각각에 대한 상태 변수가  $(n, k_1, k_2)$  및  $(m, k'_1, k'_2)$ 라고 할 때, 각 목적함수에 대한 비용은 다음과 같이 순차적으로 계산될 수 있다.

$$Makespan(s') = \begin{cases} ETA_1^R(a_{k'_m}^m), & \text{if } k_1 = k_2 = 0 \\ \max[ETA_1^R(a_{k'_m}^m), Makespan(s) + sep(a_{k_n}^n, a_{k'_m}^m)], & \text{otherwise} \end{cases} \quad (5)$$

$$TotalDelay(s') = \begin{cases} 0, & \text{if } k_1 = k_2 = 0 \\ TotalDelay(s) \\ + [Makespan(s') - ETA_1^R(a_{k_m}^m)], & \text{otherwise} \end{cases} \quad (6)$$

여기서  $a_{k_n}^n$ 은 이전 시퀀스  $s$ 의 마지막 항공기이고  $a_{k_m}^m$ 은 시퀀스  $s'$ 의 마지막 항공기이며,  $sep(a_{k_n}^n, a_{k_m}^m)$ 은  $a_{k_n}^n$ 와  $a_{k_m}^m$  간의 분리 간격을 나타낸다.

앞서 정의된 상태변수 및 비용함수에 의하여 MODP 스케줄링은 Fig. 3와 같이 총 여섯 단계로 이루어진다.

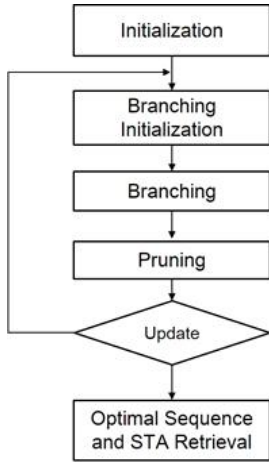


Fig. 3 MODP scheduling algorithm flow chart

1) Initialization

이 단계에서는 최종적인 항공기 시퀀스 해가 저장될 집합  $\Omega$ 에 항공기가 포함되지 않은 시퀀스  $s_\emptyset$ 를 저장하고,  $\Omega$ 의 시퀀스에 대한 계획 도착 시각(STA)이 저장될 집합  $T$ 에  $s_\emptyset$ 에 대한  $STA\ t(s_\emptyset) = \{\emptyset\}$ 를 포함시킨다. 또한 branching stage를 나타내는 변수  $q$ 를 0으로 초기화한다.

2) Branching Initialization

이 단계에서는 현재 stage의 항공기 시퀀스 해가 저장될 집합  $\tilde{\Omega}_q$  및 해당 항공기 시퀀스에 대한 STA가 저장될  $\tilde{T}_q$ 를 공집합으로 초기화한다. 또한 stage를 나타내는 변수  $q$ 을 1만큼 증가시킨다.

3) Branching

이 단계에서는  $\Omega$ 에 포함된 모든 시퀀스  $s$ 에 대해서 전체 스케줄링 대상 항공기 중 아직  $s$ 에 포함되지 않았고,  $s$ 에 포함된 항공기 중 같은 sequencing leg를 통해 도착하는 모든 항공기보다  $ETA_1^R$ 이 낮은 항공기를 추가한 시퀀스  $s'$ 를  $\tilde{\Omega}_q$ 에 저장한다. 그리고 추가된 시퀀스  $s'$ 에 대한 비용  $Makespan(s')$  및  $TotalDelay(s')$ 를 식 (5) 및 (6)을 통해 재귀적으로 계산한다.

비용들이 계산되고 나면,  $T$ 에 포함된 시퀀스  $s$ 에 해당하는 STA  $t(s)$ 에 추가된 항공기의 계획 도착 시각인  $Makespan(s')$ 를 추가하여 새로운 시퀀스  $s'$ 에 대한 STA  $t(s')$ 를 만든 후, 이를  $\tilde{T}_q$ 에 저장한다.

4) Pruning

이 단계에서는  $\tilde{\Omega}_q$ 에 저장된 동일한 상태변수를 갖는 시퀀스 중에서 다른 해에 비해 두 개의 목적 함수에 대해서 열등한 비용을 갖는 해를 제거한다. 즉, 두 개의 시퀀스  $s_1$  및  $s_2$ 에 대해서  $state(s_1) = state(s_2)$ 이고, 다음의 식 (7) 또는 (8)의 조건이 만족할 때  $s_2$ 는  $s_1$ 에 비해 열등하며, 해에서 제거된다.

$$Makespan(s_1) \leq Makespan(s_2) \text{ and } TotalDelay(s_1) < TotalDelay(s_2) \quad (7)$$

$$Makespan(s_1) < Makespan(s_2) \text{ and } TotalDelay(s_1) \leq TotalDelay(s_2) \quad (8)$$

또한 제거된 시퀀스에 해당하는 STA를  $\tilde{T}_q$ 에서 제거한다.

5) Update

이 단계에서는  $\Omega$ 을  $\tilde{\Omega}_q$ 으로,  $T$ 를  $\tilde{T}_q$ 으로 대체한 후,  $q$ 가 스케줄링 대상 항공기의 대수보다 작은 경우 다음 단계로 진행하고, 그렇지 않은 경우 Branching Initialization 단계로 진행하여 반복한다.

6) Optimal Sequence and STA Retrieval

이 단계에서는 해집합  $\Omega$ 에서 단일 최적해를 선택한 후 각 항공기의 최종 계획 도착순서  $s^*$  및 시각  $t(s^*)$ 를 계산하고 알고리즘을 종료한다.

### 3.4 조연 정보 생성

3.4절에서는 스케줄링 결과를 토대로 관제사 조연 정보를 생성하는 방법에 대하여 설명하였다. 조연 정보는 두 가지로, 첫 번째는 sequencing leg의 시작 지점에 도달하기 전에 처리되어야 하는 지연 시간이고, 두 번째는 계획 착륙 시각을 준수하기 위하여 direct-to point로 비행하도록 지시되어야 할 지점에 대한 정보이다. 각각에 대해서는 다음 세부 절에서 설명되었다.

#### 3.4.1 Sequencing leg 진입 전 지연 시간

Sequencing leg 진입 전에 처리되어야 할 지연 시간은 MODP 스케줄링에 따른 STA를 준수하기 위하여 Point-Merge sequencing leg를 초과하여 비행해야 하는 경우에 대한 초과 비행시간과 sequencing leg에 진입하기 전에 다른 항공기와의 충돌을 방지하기 위하여 필요한 지연시간의 합으로서, 다음과 같이 계산된다.

$$ExcessDelay(a) = FCFSDelay(a) + \max[0, STA(a) - ETA_P^R(a)] \quad (9)$$

여기서  $FCFSDelay$ 는 3.2절에서 설명된 FCFS Scheduling 결과로서 merge fix에서의 충돌 방지를 위한 지연시간이며,  $STA$ 는 3.3절에서 설명된 MODP Scheduling 결과로서 활주로 계획 도착 시각이고,  $ETA_P^R$ 은 3.1절에서 설명된 항공기가 Point-Merge 절차를 통해 가장 늦게 착륙할 수 있는 시각이다.

#### 3.4.2 Direct-to 지시 지점 조연정보

Direct-to 지시 지점 조연정보는 항공기가 direct-to point로 비행하기 전 마지막으로 지나는 sequencing leg 상의 fix인  $fix^*$ 와, 해당 fix로부터 direct-to 지시를 받기까지 추가적으로 비행해야 하는 거리인  $d^*$ 로 구성된다. 이러한 조연 정보를 생성하기 위해서는 먼저 해당 항공기가 Point-Merge sequencing leg 중 어느 위치에서 합류 지점으로 direct-to 비행을 수행해야 하는지가 결정되어야 한다.

Sequencing leg 상의 fix에 대한 집합  $FIX = [fix_1, fix_2, \dots, fix_P]$ 와 각 fix에 대한  $x$  및  $y$  좌표

벡터  $X = [x_1, x_2, \dots, x_P]$  및  $Y = [y_1, y_2, \dots, y_P]$ 가 주어졌을 때, 특정 항공기  $a$ 가 해당 항공기의 계획 도착시각  $STA(a)$ 를 준수하기 위해 direct-to 비행을 시작해야 할 지점의  $x$  및  $y$  좌표  $x^*$  및  $y^*$ 는 식 (10) 및 (11)과 같이 선형 보간법을 통해 계산된다. 여기서  $P$ 는 sequencing leg 상에 사전에 정의된 fix의 총 개수이다.  $ETA^R$  벡터의 원소 중  $STA(a)$ 보다 작거나 같은 값을 가지는 원소의 개수가  $u$ 라고 할 때,  $fix^*$  및  $d^*$ 는 식 (12) 및 (13)과 같이 결정된다.

$$x^*(a) = \begin{cases} x_1 + (x_2 - x_1) \frac{STA(a) - ETA_1^R(a)}{ETA_2^R(a) - ETA_1^R(a)} & \text{if } ETA_1^R(a) \leq STA(a) < ETA_2^R(a) \\ \vdots \\ x_{P-1} + (x_P - x_{P-1}) \frac{STA(a) - ETA_{P-1}^R(a)}{ETA_P^R(a) - ETA_{P-1}^R(a)} & \text{if } ETA_{P-1}^R(a) \leq STA(a) < ETA_P^R(a) \\ x_P & \text{otherwise} \end{cases} \quad (10)$$

$$y^*(a) = \begin{cases} y_1 + (y_2 - y_1) \frac{STA(a) - ETA_1^R(a)}{ETA_2^R(a) - ETA_1^R(a)} & \text{if } ETA_1^R(a) \leq STA(a) < ETA_2^R(a) \\ \vdots \\ y_{P-1} + (y_P - y_{P-1}) \frac{STA(a) - ETA_{P-1}^R(a)}{ETA_P^R(a) - ETA_{P-1}^R(a)} & \text{if } ETA_{P-1}^R(a) \leq STA(a) < ETA_P^R(a) \\ y_P & \text{otherwise} \end{cases} \quad (11)$$

$$fix^* = fix_u \quad (12)$$

$$d^* = \sqrt{(x^* - x_u)^2 + (y^* - y_u)^2} \quad (13)$$

## 4. 시뮬레이션 결과

본 연구에서는 제주 국제공항 활주로 07을 통해 착륙하기 위한 도착 절차를 대상으로 시뮬레이션을 수행하였다. Fig. 4에는 제주 국제공항

07방향 활주로를 사용하는 경우에 대한 도착 절차가 나타나 있다[8]. 그림에 나타난 바와 같이 제주 국제공항으로 입항하는 경로는 총 6개로 각각 DOTOL, MAKET, TAMNA, TOSAN, SOSDO, TOLIS fix를 통해 접근관제구역으로 진입한다. 이 중 DOTOL을 통해 접근하는 항공기들은 CHANY fix에 진입하여 Point-Merge sequencing leg를 7000피트로 비행하며, 나머지 다섯 개의 fix를 통해 접근하는 항공기들은 CHANY fix에서 9000피트로 합류하여 sequencing leg를 비행한다. 또한 선행 항공기가 존재하지 않아 sequencing leg를 비행할 필요가 없는 경우 그림에 표시된 쇄선의 경로를 따라 direct-to point로 비행한다.

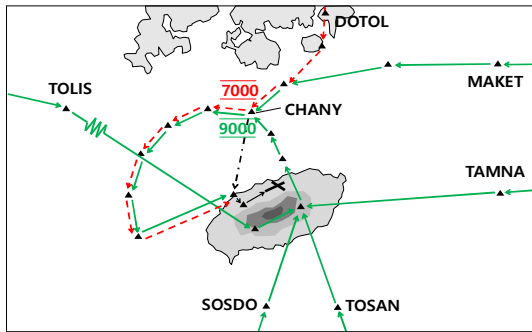


Fig. 4 Point-Merge based Standard Terminal Arrival Procedures for Jeju airport runway 07

제안된 스케줄링 알고리즘의 성능을 검증하기 위해 FCFS 기반의 스케줄링 결과와의 비교 검증을 수행하였다. FCFS 기반의 스케줄링 시에는 제안된 스케줄링 알고리즘과는 달리 CHANY fix로의 ETA가 빠른 항공기 순서대로 착륙하도록 하였다. 두 스케줄링 알고리즘의 비교 지표는 total delay로 설정하였으며, 이에 따라 MODP 스케줄링 시에는 최종적으로 계산된 최적 해집합 중 total delay를 최소화하는 스케줄을 선택하였다.

시뮬레이션은 Matlab 프로그램을 통해 수행되었으며, 항공기 기동은 BADA (Base of Aircraft Data)[9]를 기반으로 한 3 자유도 질점 모형을 기반으로 구현하였다. 또한 제주 국제공항에서 사용되는 항공기 등급별 활주로 분리 기준은 Table 1과 같이 설정하였으며, 공중에서 사용되는 레이더 분리 기준은 5마일을 적용하였다.

Table 1. Runway separation requirements

후행 \ 선행	Medium	Heavy
Medium	100 초	90 초
Heavy	160 초	120 초

항공기 대수에 따른 성능 지표 및 계산 시간을 확인하기 위하여 항공기 대수를 15대부터 24대까지 한 대 씩 증가시키며 시뮬레이션을 수행하였으며, 각 항공기 대수 별 50개의 무작위 시나리오를 생성하였다. 무작위 시나리오 생성 시 각 항공기의 접근 구역 진입 시점은 40분 시간 반경 이내에 균등 분포를 통해 설정하였으며, 각 항공기의 항공기 등급 분포 및 도착 절차는 다항 분포를 통해 결정되었다. 이 때, 항공기 등급 분포는 Medium (80%), Heavy (20%)로 설정하였으며, 도착 절차 별 교통량 분포는 DOTOL (63.5%), MAKET (19%), TAMNA (4%), TOSAN (4%), SOSDO (5.5%), TOLIS (4%)로 가정하였다.

Fig. 5에는 항공기 별 평균 지연시간이 나타나 있으며, 각 결과는 50개의 무작위 시나리오에 대한 평균값을 의미한다. 그림의 검은색 그래프는 FCFS 기반의 스케줄을 적용했을 때 항공기 한 대당 평균 지연시간을 나타내며, 흰색 그래프는 제안된 방식의 스케줄링 결과를 적용했을 때 항공기 한 대당 평균 지연시간을 나타낸다. 시뮬레이션 결과 FCFS 기반의 스케줄 대비 제안된 스케줄링 알고리즘 결과를 적용 시 약 10~15% 지연 시간을 감소시킬 수 있을 것으로 나타났다.

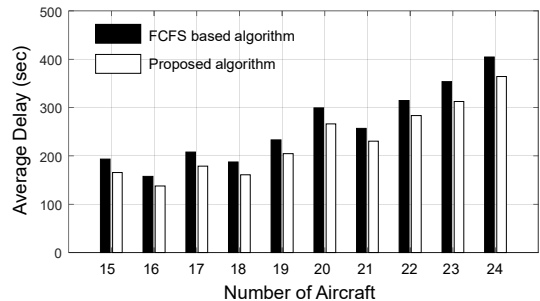


Fig. 5 Simulation results (average delay)

Fig. 6에는 항공기 대수에 따른 스케줄링 계산 소요시간이 box-and-whisker 도표 형태로 나타나 있다. 그래프에 나타나 있는 박스는 25, 50, 75 백분위수 값을 의미한다. 또한 '+' 기호는 outlier

값을 나타내며, whisker 선은 outlier를 제외한 극단값을 나타낸다. Fig. 6을 통해 본 논문에서 제안된 스케줄링 알고리즘은 24대 이하의 항공기에 대한 스케줄링을 수행하는데 최대 0.16초가 소요되어, 실시간으로 사용되기에 적절함을 확인할 수 있다.

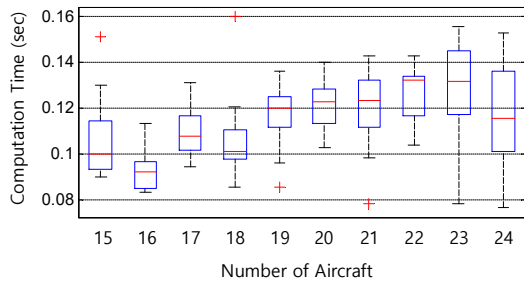


Fig. 6 Simulation results (computation time)

## 5. 결론

본 연구에서는 Point-Merge 절차 기반의 도착 관리 시스템에 사용될 수 있는 스케줄링 알고리즘을 제안하였다. 제안된 알고리즘은 FCFS 및 MODP 기반의 스케줄링 방식을 단계적으로 사용함으로써 동일한 절차 내에서의 항공기 간 추월을 방지함과 동시에 항공기의 순서가 Point-Merge 절차의 sequencing leg를 통해 효율적으로 결정될 수 있도록 하였다.

향후에는 알고리즘을 실제 상황에서 사용하기 위해서 불확실성을 고려할 수 있는 방법을 연구해 봄과 동시에 Human-In-The-Loop 시뮬레이션 등 더욱 신뢰성이 높은 실험을 통해 알고리즘을 검증해 볼 예정이다. 본 연구는 스케줄링을 통해 원활한 항공교통흐름을 가능하게 함으로써 항공기 운항을 효율화할 수 있으며, 이로 인하여 항공기의 온실 가스 및 연료 소모량 감소에 기여할 수 있다.

## 후 기

본 논문은 한국항공운항학회 2017년 춘계학술대회에 발표된 논문을 수정 보완한 논문이며 본 연구는 국토교통부 연구과제인 ‘항공기 출발 및 도착 통합 관리 기술 연구’에 의해 연구비 지원을 받았음.

## Reference

- [1] Erzberger, H., and Nedell, W., “Design of Automated System for Management of Arrival Traffic,” NASA TM-102201, 1989.
- [2] Eurocontrol, “AMAN Status Review 2010,” 2010.
- [3] Oh, E., Kim, H., Eun, Y. and Jeon, D., “Study on Trajectory Prediction Accuracy Analysis Method for Performance Improvement of a Trajectory Prediction Module of Arrival Manager”, Journal of Korean Society for Aviation and Aeronautics, Vol. 23, No. 3, 2015, pp. 28-34.
- [4] Liang, M., Delahaye, D. and Marechal, P., “A Framework of Point Merge-based Autonomous System for Optimizing Aircraft Scheduling in Busy TMA,” Fifth SESAR Innovation Days, 2015.
- [5] Eurocontrol, “Point Merge Integration of Arrival Flows Enabling Extensive RNAV Application and Continuous Descent - Operational Services and Environment Definition,” 2010.
- [6] Carraway, R. L. and Morin, T. L., “Theory and applications of generalized dynamic programming: an overview,” International Journal of Computers and Mathematics with Applications, 16, 779-788, 1988.
- [7] 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, 2009.
- [8] Office of Civil Aviation of Republic of Korea, Aeronautical Information Publication, RKPC AD CHART 2-18, 12 MAY 2016.
- [9] EUROCONTROL Experimental Center, “User Manual for the Base of Aircraft Data (BADA) Revision 3.10,” EEC Technical/Scientific Report No. 12/04/10-45, 2012.