

<http://dx.doi.org/10.7236/IIBC.2016.16.1.245>

IIBC 2016-1-34

## 다중공항 시스템의 도착-출발 가용량 배정 알고리즘

### Arrival-Departure Capacity Allocation Algorithm for Multi-Airport Systems

이상운\*

Sang-Un, Lee \*

**요 약** 본 논문은 다중공항의 도착/출발 문제에 대해 항공기 지연을 최소화시키는 최적 해를 얻을 수 있는 휴리스틱 알고리즘을 제안하였다. 단일 공항의 출발/도착 항공기의 지연 대수를 최소화시키는 최적 해를 찾는 문제에 대해 수학적 방법만이 제안되고 있다. 다중공항의 경우는 선형계획법이나 메타휴리스틱 방법의 일종인 유전자 알고리즘이 적용되고 있다. 제안된 알고리즘은 먼저, 각 공항의  $i$ 번째 단위시간 (15분)에서 총 도착/출발 항공기 대수에 대해 지연을 최소화시키는 운영 능력들 중 중앙값을 선택하였다. 다음으로 공항간 도착 항공기의 도착지를 변경시켰다. 실험 결과 제안된 알고리즘은 유전자 알고리즘에 비해 지연 항공기 대수에 대해 보다 좋은 결과를 얻었다.

**Abstract** This paper suggests a heuristic algorithm to obtain optimal solution of minimum number of aircraft delay in multi-airport arrivals/departures problem. This single airport arrivals/departures problem can be solved by mathematical optimization method only. The linear programming or genetic algorithm that is a kind of metaheuristic method is used for a multi-airport arrivals/departures problem. Firstly, the proposed algorithm selects the median minimum delays capacity in various arrivals/departures capacities at an airport for the number of aircraft in  $i^{th}$  time interval (15 minutes) at each airport. Next, we suggest reallocate method for arrival aircraft between airports. This algorithm better result of the number of delayed aircraft then genetic algorithm.

**Key Words** : Multi-airport, Minimization of Delays, Capacity, Median, Reallocate

## 1. 서 론

공항 활주로의 항공기 도착/출발 처리 가용량(capacity)이 항공교통 요구량(도착, 출발)을 충족시키지 못할 경우 극심한 혼잡과 더불어 항공기 지연에 따른 막대한 비용과 고객 불만족도 증가한다. 따라서 관제사는 15분 간격으로 해당 공항 활주로의 이·착륙 가용능력을

최대한 활용하여 이·착륙 지연 대수를 최소화 시키는 것이 주요 임무이다.<sup>[1]</sup>

공항의 이·착륙 최소 지연 대수  $z$  값을 찾는 문제에 대해 Idrissi[2]는 분기한정법+지역탐색법+시간 초과 제약사항법의 수학적 최적화 방법을, Gilbo[1,3]는 출발/도착 교환 최적화 도구와 수학적 최적화 방법을 적용하였다. 이와 같이 수학적 방법 이외에 구체적인 알고리즘은

\*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과  
접수일자 : 2015년 7월 3일, 수정완료 : 2016년 1월 2일  
게재확정일자 : 2016년 2월 5일

Received: 3 July, 2015 / Revised: 2 January, 2016 /

Accepted: 5 February, 2016

\*Corresponding Author: [sulee@gwnu.ac.kr](mailto:sulee@gwnu.ac.kr)

Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

제시되지 못하고 있다.[4] 또한, 단일 공항이 아닌 다중공항을 관제하는 경우 문제는 더욱 더 복잡해진다. 다중공항 관제의 경우 휴리스틱 알고리즘이 존재하지 않아 Hu et al.[5]은 선형계획법(linear programming)과 메타휴리스틱 방법의 일종인 유전자 알고리즘(genetic algorithm)을 적용하였다.

본 논문은 다중공항의 출발/도착 항공기의 지연을 최소화시키고, 공항간 도착 항공기 조정을 통해 최적 해를 찾을 수 있는 휴리스틱 알고리즘을 제안한다. 제안된 방법은 각 공항에 대해  $i$ 번째 시간간격에서 지연된 항공기 대수와 출발/도착 예정 항공기 대수를 합한 값에 대해 지연을 최소화시키는 가용 능력들 중에서 중앙값(median)을 단순히 선택하는 방법을 제안하였다. 이 결과에 대해 도착 지연을 최소화시키도록 공항간 도착 항공기를 재배정하는 방법을 제안하였다.

2장에서는 단일 공항의 가용능력을 최대화시켜 도착/출발 지연 항공기 대수를 최소화시키는 문제에 대한 개념과 더불어 다중공항에 대한 연구결과를 고찰해 본다. 3장에서는 다중공항의 최소 지연 항공기 대수를 찾는 휴리스틱 알고리즘을 제시한다. 4장에서는 Hu et al.[5]가 제시한 문제에 제안된 알고리즘을 적용하여 알고리즘의 적합성을 검증해 본다.

## II. 관련연구와 문제점

항공교통의 목표는 15분 간격으로 주어진 공항의 출발과 도착 처리능력을 최대화시켜 지연 대수를 최소화시키는 것이다. 이때 공항의 활주로는 1개 이상이며, 도착과 출발하는 항공기는 다수의 접근/이탈 경로인 픽스(fixes)를 이용한다. Fix는 선회 트랙의 시작점으로 VOR(VHF omnidirectional range)과 같은 라디오 비콘(radio beacon)을 기점으로 한다.

전형적인 공항의 출발-도착 스키마는 그림 1과 같다.[1] 만약, 도착하는 항공기 대수가 공항의 처리능력을 초과하는 경우, 다수의 항공기가 착륙을 위해 대기하는 큐(queue)가 발생한다. 착륙대기 큐는 1,000ft 고도 차이, 4 NM의 선회 트랙을 가진 다수의 픽스들로 구성되어 있다. 이륙(출발)도 동일한 방법을 적용한다.[6]

공항은 활주로 수와 바람방향에 따른 제약조건으로 인해 이륙과 착륙 전용의 활주로가 없으며, 항상 혼용하

여 운영하고 있으며, 착륙 중인 항공기를 피하여 출발 항공기가 활주로를 교차하기도 한다. 즉, 15분 동안의 공항의 처리능력은 이륙과 착륙 항공기의 대수를 교환하여 운영하고 있으며, 전형적인 교환은 그림 2에 제시되어 있다.[1,2,7]

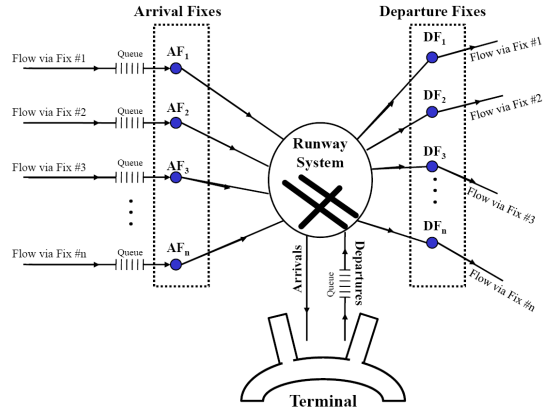


그림 1. 전형적인 도착-출발 스키마  
Fig. 1. Typical Arrival-Departure Scheme

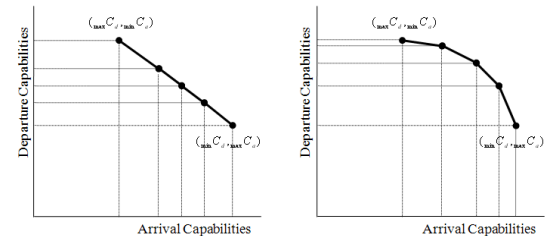


그림 2. 전형적인 도착/출발 교환 곡선  
Fig. 2. Typical Arrival/Departure Trade-off Curve

여기서 제기되는 문제점은 대부분의 경우 출발/도착 항공기 대수가 공항의 운영 능력을 초과한다는 점이다. 이로 인해 특정한 15분 동안 지연된 항공기는 다음 15분 동안의 출발/도착 항공기에도 연쇄적으로 영향을 미친다. 따라서 그림 2의 도착/출발 교환 곡선을 적절히 배분하여 지연 항공기 대수를 최소화시키는 것이 관제사에게 주어진 가장 큰 임무이다.

다중공항은 하나의 주 공항(main airport)과 다수의 인접한 위성공항(satellite airports)으로 구성된 형태이다. 주 공항은 주로 국제선 전용이며, 위성공항은 국제선 능력도 갖추고 있지만 주로 국내선 전용으로 운영된다. 예로 주 공항은 인천공항, 위성공항은 김포공항이라 할 수 있다. 이러한 다중공항을 관제하는 경우, 주 공항의 항공

교통 혼잡이 발생하면 주 공항에 도착 예정인 항공기의 일부를 위성공항으로 착륙시키도록 경로를 변경시켜야 한다. 이따금 악천후나 비상상황이 발생하거나 위성공항의 혼잡이 예상되면 계획된 도착 예정 항공기를 다른 공항으로 경로를 변경시키는 경우도 발생한다.<sup>[5]</sup>

공항의 가용능력을 최대로 배분하여 도착/출발 항공기들의 지연 대수를 최소화시키는 문제에 대해, 다음의 용어들을 적용한다.

- $i = 1, 2, \dots, m$ : 15분 간격
- $Q_{a(i-1)}$ :  $i-1$ 번째 시간 간격에서 선회비행으로 미착륙 중인 도착 지연 항공기 대수
- $Q_{d(i-1)}$ :  $i-1$ 번째 시간 간격에서 지상에서 대기중인 출발 지연 항공기 대수
- $X_{a(i)}$ :  $i$ 번째 시간 간격에서 도착 예정 항공기 대수,  $X_{a(i)} = X_{a(i)} + Q_{a(i-1)}$
- $Y_{d(i)}$ :  $i$ 번째 시간 간격에서 출발 예정 항공기 대수,  $Y_{d(i)} = Y_{d(i)} + Q_{d(i-1)}$
- $C_j = (C_{a(j)}, C_{d(j)})$ ,  $j = 1, 2, \dots, n$ : 공항의 가용능력
- $X_{aa(i)}$ : 공항의 착륙 가용능력  $C_{a(j)}$ 를  $i$ 번째 시간 간격에서 할당한 값 (착륙 가능 항공기 대수)
- $Y_{ad(i)}$ : 공항의 이륙 가용능력  $C_{d(j)}$ 를  $i$ 번째 시간 간격에서 할당한 값 (이륙 가능 항공기 대수)
- $Q_{a(i)}$ :  $i$ 번째 시간 간격에서 미착륙 중인 도착 지연 항공기 대수,  $Q_{a(i)} = X_{a(i)} - X_{aa(i)} > 0$
- $Q_{d(i)}$ :  $i$ 번째 시간 간격에서 지상에서 대기중인 이륙 지연 항공기 대수,  $Q_{d(i)} = Y_{d(i)} - Y_{ad(i)} > 0$
- $z_{1,2}$ : 공항 #1에서 공항 #2로의 경로 변경 항공기 대수
- $z_{2,1}$ : 공항 #2에서 공항 #1으로의 경로 변경 항공기 대수

표 1은 Hu et al.<sup>[5]</sup>가 제시한 2개 공항 (주 공항과 위성 공항)의 3시간 주기에 대한 항공교통량과 도착/출발 교환 곡선인 공항 처리 능력을 보여주고 있다. 즉,  $X_{a(i)}$ ,  $Y_{d(i)}$ ,  $i = 1, 2, \dots, m$ 가 주어졌을 때,  $C_{a(j)}$ ,  $C_{d(j)}$ ,  $j = 1, 2, \dots, n$  중에서 어떤 가용량을 배정하여 지연을 최소화 시키느냐가 가용량 할당 문제이다.

표 1. 다중공항의 항공 교통량과 공항 능력  
 Table 1. Air Traffic and Airport Capacity for Multi-airport

(a) 공항 #1

|       |    |    |    |    |   |    |    |    |    |    |    |    |     |
|-------|----|----|----|----|---|----|----|----|----|----|----|----|-----|
| $i$   | 1  | 2  | 3  | 4  | 5 | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 계   |
| $X_a$ | 26 | 38 | 42 | 29 | 6 | 13 | 14 | 20 | 40 | 25 | 13 | 12 | 278 |
| $Y_d$ | 36 | 32 | 9  | 15 | 7 | 10 | 17 | 33 | 34 | 22 | 13 | 1  | 229 |

|       |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|
| $j$   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 |
| $C_a$ | 17 | 18 | 19 | 20 | 21 | 22 | 24 | 25 | 26 | 27 | 28 |
| $C_d$ | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 21 | 19 | 17 | 15 |
| 계     | 47 | 47 | 47 | 47 | 47 | 47 | 48 | 46 | 45 | 44 | 43 |

(b) 공항 #2

|       |    |    |   |    |    |   |    |    |   |    |    |    |     |
|-------|----|----|---|----|----|---|----|----|---|----|----|----|-----|
| $i$   | 1  | 2  | 3 | 4  | 5  | 6 | 7  | 8  | 9 | 10 | 11 | 12 | 계   |
| $X_a$ | 20 | 21 | 4 | 29 | 30 | 8 | 15 | 11 | 9 | 19 | 30 | 8  | 204 |
| $Y_d$ | 27 | 11 | 4 | 24 | 5  | 7 | 25 | 13 | 7 | 17 | 25 | 1  | 166 |

|       |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|
| $j$   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 |
| $C_a$ | 14 | 15 | 16 | 18 | 19 | 21 | 22 | 23 | 24 | -  | -  |
| $C_d$ | 26 | 25 | 24 | 23 | 22 | 21 | 18 | 15 | 12 | -  | -  |
| 계     | 40 | 40 | 40 | 41 | 41 | 42 | 40 | 38 | 36 | -  | -  |

표 2에는 표 1에 대해 Hu et al.<sup>[5]</sup>이 제시한 RHC\_LP의 결과를 제시하였다. 여기서, RHC는 N-step ahead online optimization strategy, LP는 선형계획법, GA는 유전자 알고리즘을 의미한다.

표 2. RHC\_GA와 RHC\_LP 알고리즘 결과  
 Table 2. Result of RHC\_GA and RHC\_LP Algorithm

| 시간<br>간격 | 공항 #1 |           |       |       |       | 공항 #2 |           |       |       |       |
|----------|-------|-----------|-------|-------|-------|-------|-----------|-------|-------|-------|
|          | 도착    |           |       | 출발    |       | 도착    |           |       | 출발    |       |
|          | $C_a$ | $z_{2,1}$ | $Q_a$ | $C_d$ | $Q_d$ | $C_a$ | $z_{1,2}$ | $Q_a$ | $C_d$ | $Q_d$ |
| 1        | 19    | 0         | 7     | 28    | 8     | 19    | 0         | 1     | 22    | 5     |
| 2        | 24    | 0         | 21    | 24    | 16    | 22    | 0         | 0     | 18    | 0     |
| 3        | 25    | 0         | 17    | 21    | 4     | 23    | 21        | 2     | 15    | 0     |
| 4        | 25    | 0         | 21    | 21    | 0     | 21    | 0         | 10    | 21    | 3     |
| 5        | 26    | 0         | 1     | 19    | 0     | 24    | 0         | 16    | 12    | 0     |
| 6        | 24    | 0         | 0     | 24    | 0     | 24    | 0         | 0     | 12    | 0     |
| 7        | 21    | 0         | 0     | 26    | 0     | 15    | 0         | 0     | 25    | 0     |
| 8        | 17    | 0         | 0     | 30    | 3     | 21    | 3         | 0     | 21    | 0     |
| 9        | 17    | 0         | 9     | 30    | 7     | 23    | 14        | 0     | 15    | 0     |
| 10       | 24    | 0         | 10    | 24    | 5     | 19    | 0         | 0     | 22    | 0     |
| 11       | 26    | 1         | 0     | 19    | 0     | 21    | 0         | 8     | 21    | 4     |
| 12       | 24    | 0         | 0     | 24    | 0     | 21    | 0         | 0     | 21    | 0     |
| 계        | 272   | 1         | 86    | 290   | 43    | 253   | 38        | 37    | 225   | 12    |
| RHC_LP   | 278   | -         | 143   | 229   | 77    | 204   | -         | 31    | 166   | 12    |

Gilbo<sup>[3]</sup>는 단일 공항의 지연 항공기 최소화 전략으로 출발 항공기 지연 최소화, 균형, 도착 항공기 지연 최소화의 3가지 전략을 제안하고,  $\alpha = [0, 1]$ 을 고려하여 식 (1)을 찾고자 하였다. 여기서,  $\alpha = 1$ 은 도착 지연 최소화를,  $\alpha = 0$ 이면 출발 지연 최소화를,  $\alpha = 0.5$ 이면 도착과 출발에 동일한 우선순위를 부여하여 우선순위가 없는 상태를

의미한다.

$$\min \sum_{i=1}^n [\alpha X_{i+1} + (1-\alpha) Y_{i+1}], 0 \leq \alpha \leq 1 \quad (1)$$

기존의 방법 대부분은  $z$  값 여부에 상관없이  $\alpha=0.5$ 와  $\alpha=0.7$ 에 대해 수학적 방법들을 제안하고 있다.  $\alpha=0.5$ 인 경우 대부분  $z$  값을 만족하지만  $\alpha=0.7$ 인 경우  $z$  값을 초과하는 경우가 대부분이다. 이 3가지 전략 중 어떤 방법이 최적인지에 대한 기준은 아직 설정되어 있지 않고 있다.<sup>[3]</sup>

Idrissi<sup>[2]</sup>와 Gilbo<sup>[1,3]</sup>는 수학적 방법으로  $z$  값을 찾았다. 그러나  $z$ 를 찾을 수 있는 휴리스틱 알고리즘은 아직까지 제안되지 않고 있다.

또한, 공항의 활주로 배정문제에 대해서는 Lee<sup>[8]</sup>이, 단일공항의 도착-출발에 대한 지연 최소화에 대해서는 Lee<sup>[9]</sup>가 연구하였지만 다중공항의 도착-출발 지연 최소화에 대해서는 연구가 수행되지 않고 있다.

### III. 다중공항의 도착 지연 최소화 알고리즘

본 장에서는 먼저, 도착/출발 최소 지연 항공기 대수인  $z$  값을 찾는 휴리스틱 알고리즘을 제안하고, 다음으로, 도착 지연 최소화를 위해 다중공항에 도착 항공기를 분산 배치하는 방법을 제안한다.

첫 번째로, 각 공항의 도착/출발 최소 지연 항공기 대수를 구하기 위해 공항의 가용량을 배정하는 방법을 제안한다. 제안된 알고리즘은 시간 간격  $i=1,2,\dots,m$ 에서 공항 가용능력  $C_j, j=1,2,\dots,n$ 에 대한 출발/도착 지연 항공기 대수  $Q_j$ 를 구하여 최소 지연 대수  $\min Q_j$ 를 가진 범위  $[C_j=L, C_j=H]$ 를 결정한다.  $Q_j$ 는 다음과 같이 계산된다.

$$Q_{a(j)} = \begin{cases} 0 & , X_{a(i)} - C_{a(j)} \leq 0 \\ X_{a(i)} - C_{a(j)} & , X_{a(i)} - C_{a(j)} > 0 \end{cases}$$

$$Q_{d(j)} = \begin{cases} 0 & , Y_{d(i)} - C_{d(j)} \leq 0 \\ Y_{d(i)} - C_{d(j)} & , Y_{d(i)} - C_{d(j)} > 0 \end{cases}$$

$$Q_j = Q_{a(j)} + Q_{d(j)}, j=1,2,\dots,n$$

다음으로,  $\min Q_j = [C_j=L, C_j=H]$ 에 대해  $i=1,2,\dots,m$

순서대로 중앙값 선택법 (median method, MM)으로 공항 가용량  $C_j$ 를 배정한다.

if  $Q_{a(i-1)} = 0, Q_{d(i-1)} = 0, (H-L) = 1, k=2,$  and  $C_j > C_{j+1}$  then  $C_j$  선택  
else

if  $(H-L) = \text{odd}, k = \text{even}$  then

if  $X_{a(i)} > Y_{d(i)}$  then  $C_j, j = \left\lfloor \frac{H-L}{2} \right\rfloor$  선택

else if  $X_{a(i)} < Y_{d(i)}$  then  $C_j, j = \left\lceil \frac{H-L}{2} \right\rceil$  선택

else if  $(H-L) = \text{even}, k = \text{odd}$

then  $C_j, j = (H-L)/2$  선택.

$X_{aa(i)} \leftarrow C_{a(j)}, Y_{ad(i)} \leftarrow C_{d(j)}$  배정.

$Q_{MED} = \sum_{i=1}^m (Q_{a(i)} + Q_{d(i)})$  계산.

두 번째로, 도착 항공기 지연 최소화를 위해 다중공항의 도착항공기 분산 배치 방법은  $z_{1,2}, z_{2,1}$  순서로 다음 조건을 만족시킬 때까지 최적화(재배정) 과정을 수행한다.

$(Q_{a1} > Q_{a2}) \cap Q_{a1} > 0$ 에 대해

$z = (X_{aa2} - Q_{ad(i-1)} + X_{ad(i)}) > 0$  계산,  $z_{1,2} = \min\{Q_{a1}, z\}$ 로 결정

$X_{a1} \leftarrow X_{a1} - z_{1,2}, X_{a2} \leftarrow X_{a2} + z_{1,2}$

공항 #1과 #2에 대해 MM으로  $C_j$  조정.

$(Q_{a2} > Q_{a1}) \cap Q_{a2} > 0$ 에 대해

$z = (X_{aa1} - Q_{ad(i-1)} + X_{ad(i)}) > 0$  계산,  $z_{2,1} = \min\{Q_{a2}, z\}$ 로 결정

$X_{a2} \leftarrow X_{a2} - z_{2,1}, X_{a1} \leftarrow X_{a1} + z_{2,1}$

공항 #1과 #2에 대해 MM으로  $C_j$  조정.

제안된 알고리즘을 도착지연 최소화(minimum arrival delay, MAD) 알고리즘이라 하자.

### IV. 적용 결과 및 분석

표 1의 데이터에 대해 제안된 MAD 알고리즘을 적용하여 각 공항의 가용량을 배정한 결과는 표 3에 제시되어 있다. 표 4에는 MAD 알고리즘으로 공항간 도착 항공기



표 5. MAD 알고리즘의 최적해

Table 5. The Optimal Solution of MAD Algorithm

| i  | z <sub>1,2</sub> |          |          |          |    | z <sub>2,1</sub> |          |          |    | Airport #1       |                  |                |                | Airport #2       |                  |                |                |
|----|------------------|----------|----------|----------|----|------------------|----------|----------|----|------------------|------------------|----------------|----------------|------------------|------------------|----------------|----------------|
|    | Trial #1         | Trial #2 | Trial #3 | Trial #4 | 계  | Trial #1         | Trial #2 | Trial #3 | 계  | X <sub>arr</sub> | Y <sub>arr</sub> | Q <sub>a</sub> | Q <sub>d</sub> | X <sub>arr</sub> | Y <sub>arr</sub> | Q <sub>a</sub> | Q <sub>d</sub> |
| 1  | -                | -        | -        | -        | 0  | -                | -        | -        | -  | 24               | 24               | 2              | 12             | 19               | 22               | 1              | 5              |
| 2  | -                | -        | -        | -        | 0  | -                | -        | -        | -  | 24               | 24               | 16             | 20             | 22               | 18               | 0              | 0              |
| 3  | 15               | 3        | 1        | 1        | 20 | -                | -        | -        | -  | 24               | 24               | 14             | 5              | 24               | 12               | 0              | 0              |
| 4  | -                | -        | -        | -        | 0  | -                | -        | -        | -  | 26               | 19               | 17             | 1              | 21               | 21               | 8              | 3              |
| 5  | -                | -        | -        | -        | 0  | 3                | 3        | 1        | 7  | 28               | 15               | 0              | 0              | 24               | 12               | 9              | 0              |
| 6  | -                | -        | -        | -        | 0  | -                | -        | -        | -  | 22               | 25               | 0              | 0              | 22               | 18               | 0              | 0              |
| 7  | -                | -        | -        | -        | 0  | -                | -        | -        | -  | 22               | 25               | 0              | 0              | 15               | 25               | 0              | 0              |
| 8  | -                | 1        | -        | 1        | 2  | -                | -        | -        | -  | 18               | 29               | 0              | 4              | 19               | 22               | 0              | 0              |
| 9  | 10               | 3        | 1        | 1        | 15 | -                | -        | -        | -  | 24               | 24               | 1              | 14             | 24               | 12               | 0              | 0              |
| 10 | 3                | -        | -        | -        | 3  | -                | -        | -        | -  | 20               | 27               | 3              | 9              | 22               | 18               | 0              | 0              |
| 11 | -                | -        | -        | -        | 0  | 4                | 4        | 2        | 10 | 24               | 24               | 0              | 0              | 21               | 21               | 1              | 4              |
| 12 | -                | -        | -        | -        | 0  | -                | -        | -        | -  | 22               | 25               | 0              | 0              | 22               | 18               | 0              | 0              |
| 계  |                  |          |          |          | 40 |                  |          |          | 17 |                  |                  | 53             | 65             |                  |                  | 19             | 12             |

표 6. 알고리즘 비교

Table 6. Compare of Algorithms

| 모델     | Q <sub>a</sub> | Q <sub>d</sub> | Q = Q <sub>a</sub> + Q <sub>d</sub> |
|--------|----------------|----------------|-------------------------------------|
| RHC_LP | 174            | 89             | 263                                 |
| RHC_GA | 123            | 55             | 178                                 |
| CDO_GA | 167            | 63             | 230                                 |
| MAD    | 72             | 77             | 149                                 |

경로를 변경시킨 최종 결과를 제시하였다. 여기서는 공항 #1에서 공항 #2로의 5회 경로 변경과 공항 #2에서 공항 #1로의 4회 경로변경 과정이 수행되었다. 표 4의 결과를 요약하여 표 5에 제시하였다.

표 6은 Hu et al.<sup>[5]</sup>의 결과와 MAD 알고리즘의 항공기 지연대수를 비교한 결과이다. 표로부터 제안된 MAD 알고리즘이 지연 항공기 대수를 최소화 시키도록 도착지 변경을 통해 다중공항의 가용량을 배정하였음을 알 수 있다.

## V. 결론

본 논문은 아직까지 수학적 최적화 방법 이외에는 휴리스틱 알고리즘이 존재하지 않고 있는 다중공항의 출발/도착 항공기의 지연 최소화 z를 찾는 휴리스틱 알고리즘을 제안하였다.

제안된 알고리즘은 i번째 단위 시간 (15분 간격)에서 지연되고 있는 항공기 대수를 포함한 출발/도착 예정 항공기 대수에 대해 지연을 최소화 시키는  $k \leq j$ 개

$$Q_a = 53 + 19 = 72, Q_d = 65 + 12 = 77$$

의 공항 운영 능력들 중 중앙값을 선택하는 방법을 적용하였다. 또한, 공항 간 항공기의 도착지를 변경시키는 방법을 통해 다중공항의 지연 항공기 대수를 최소화시켰다.

제안 알고리즘은 기존의 메타휴리스틱 방법의 일종인 유전자 알고리즘에 비해 좋은 결과를 얻었다.

## References

- [1] E. P. Gilbo, "Optimizing Airport Capacity Utilization in Air Traffic Flow Management Subject to Constraints at Arrival and Departure Fixes," IEEE Trans. on Control Systems Technology, Vol. 5, No. 5, pp. 490-503, doi:10.1109/87.623035, Sep. 1997.
- [2] A. Idrissi, "Some Methods to Treat Capacity Allocation Problems," Journal of Theoretical and Applied Information Technology, Vol. 37, No. 2, pp. 141-158, Mar. 2012.
- [3] E. P. Gilbo, "Arrival/Departure Capacity Tradeoff Optimization: A Case Study at the St. Louis Lambert International Airport (STL)," 5th USA/Europe Air Traffic Management R&D Seminar, pp. 1-10, Budapest, 23-27, Jun. 2003.
- [4] A. Agustín, A. Alonso-Ayuso, L. F. Escudero, and C. Pizarro, "Mathematical Optimization Models for Air Traffic Flow Management: A Review," Studia Informatica Universalis, pp. 141-184, Feb. 2010.
- [5] X. B. Hu, W. H. Chen, and E. D. Paolo,

- "Multiairport Capacity Management: Genetic Algorithm with Receding Horizon," IEEE Trans. on Intelligent Transportation Systems, Vol. 8, No. 2, pp. 254-263, doi:10.1109/ TITS.2006.890067, Jun. 2007.
- [6] Wikipedia, "Holding (Aviation)," Wikipedia Foundation, Inc., 2015.
- [7] O. Gluchsenko, "Dynamic Usage of Capacity for Arrivals and Departures in Queue Minimization," IEEE International Conference on Control Applications (CCA), pp. 139-146, Denver, CO, USA, doi:10.1109/CCA.2011.6 044451, 2011.
- [8] S. U. Lee, "Exact Cover Algorithm for Runway Assignment Problem," Journal of KIIT, Vol. 12, No. 10, pp. 163-172, doi:10.14801/kitr.2014.12.10.163, Oct. 2014.
- [9] S. U. Lee, "An Algorithm for Capacity Allocation Problem (CAP) in Airport Arrivals-Departures System," Journal of KIIT, Vol. 13, No. 11, pp. 61-72, doi:10.14801/jkit.2015.13.11.61, Nov. 2015.

## 저자 소개

### 이 상 운(정회원)



- 1987년 : 한국항공대학교 항공전자공학과 (학사)
  - 1997년 : 경상대학교 컴퓨터과학과 (석사)
  - 2001년 : 경상대학교 컴퓨터과학과 (박사)
  - 2003년 : 강원도립대학 컴퓨터응용과 전임강사
  - 2004년 ~ 2007.2 : 국립 원주대학 여성교양과 조교수
  - 2007년 3월 ~ 2015년 3월 : 강릉원주대학교 멀티미디어공학과 부교수
  - 2015년 4월 ~ 현재 : 강릉원주대학교 멀티미디어공학과 정교수
- <주관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 그래프 알고리즘>
- E-Mail : sulee@gwmu.ac.kr