

Original Article

노드링크 모델 기반 항공기 공항 지상이동 경로 추출 기법에 대한 연구

정명숙*, 은연주*, 김현경*, 전대근*

A Study on Taxi Route Extraction Based on a Node-Link Model for Aircraft Movements on Airport Surface

Myeongsook Jeong*, Yeonju Eun*, Hyounkyoung Kim*, Daekeun Jeon*

ABSTRACT

Estimation of the taxi-out and taxi-in times of aircraft on a airport surface is one of the essential features of Departure Manager (DMAN). Especially for an airport with multiple runways and large ramp areas, estimation of the taxi-out and taxi-in times are mainly dependent on the taxi routes on airport surface. This paper described the method of automatic extraction of the the taxi routes using the ASDE track data and the Dijkstra algorithm based on the node-link model of a airport surface movements. In addition, we analyzed the ground operation status of Incheon International Airport using the extracted taxi routes.

Key Words : DMAN(항공기 출발관리시스템), Taxi Route(지상이동 경로), Node-Link Model(노드링크 모델), ASDE(공항지상감시장비)

1. 서 론

국제민간항공기구(International Civil Aviation Organization, ICAO)는 지속적으로 증가하는 항공 교통량에 대비하여 미래 항공교통 시스템 환경 구축 가이드라인(Aviation System Block Upgrades, ASBU)을 마련하고 전 세계적인 이행을 촉구하고 있다[1]. 이에 국내외를 비롯한 여러 국가들에서 자동화 및 효율성이 향상된 항공 교통 관제시스템을 개발하기 위해 다양한 기술 개

발 연구를 진행하고 있으며, 항공기 출발관리시스템(Departure Manager, DMAN) 또한 이러한 일환으로 국내 연구진에 의해 2014년부터 개발 진행 중에 있다[2].

항공기의 지상이동시간 예측은 DMAN의 핵심 기능 중 하나이며, 예측된 항공기 지상이동시간은 DMAN의 핵심 모듈인 출발 스케줄링의 근간이 되는 정보이기도 하다. 항공기의 지상이동시간은 주로 항공기 지상이동 경로(taxi route)에 따라 좌우되는데, 특히 다수 활주로를 사용하고 게이트 수가 많은 공항의 경우 출도착 활주로 및 항공기의 출도착 게이트의 위치에 따라 지상이동 경로가 다르고 주기장내 상황에 따라서도 경로가 변경되므로, 사전에 항공기 지상이동 경로를 자세히 모델링 하는 것이 항공기의 지상이동시간 예측을 위해 꼭 필요한 과정중의 하나로 볼 수

Received : 24. Aug. 2017. Revised : 11. Sep. 2017.

Accepted : 23. Sep. 2017

* 한국항공공주우연구원 차세대항행팀

연락처 E-mail : msjeong@kari.re.kr

대전시 유성구 과학로 169-84

있다. 항공기 지상이동 경로는 공항 항공정보간행물(Aeronautical Information Publication, AIP) 상에 간략히 기술된 항공기 출발 및 도착 절차를 이용하여 생성할 수도 있으나, 그럴 경우 생성된 지상이동 경로가 매우 단조로울 수 있다. 즉 공항 지상의 교통 혼잡도 및 항공기 간 충돌 등을 고려하여 실제 관제상황에서 발생할 수 있는 다양한 항공기 우회 지상이동 경로들을 항공기 지상이동시간 예측에 반영하기가 어렵다.

따라서 본 논문에서는 공항에서 실제 사용되고 있는 항공기 지상이동 경로를 보다 손쉽게 생성할 수 있도록 공항의 상세한 지상교통 모델인 노드링크(Node-Link) 모델을 기반으로 하여, 실제 항공기들이 움직인 궤적정보를 담고 있는 공항지상감시장비(ASDE, Airport Surface Detection Equipment) 항적 데이터와 Dijkstra 알고리즘을 이용하여 항공기 지상이동 경로를 자동으로 추출하는 방법에 대해 기술하였다. 또한 인천국제공항(이하 인천공항)을 대상으로 해당 방법을 적용하여 추출된 지상이동 경로를 바탕으로 인천공항의 항공기 지상이동 운영 현황을 분석하였다.

II. 본 론

1. 노드링크 모델

노드링크는 도로 교통망, 통신망 등을 네트워크 모형으로 표현하기 위한 모델링 기법으로, 본 논문에서의 노드링크 모델은 공항 지상의 상세한 이동경로 네트워크 모형을 의미한다. 즉 공항 내 항공기가 이동하는 계류장(Ramp), 유도로(Taxiway), 활주로(Runway) 등을 일정한 규칙에 따라 마디형태로 구분한 것으로, 노드는 교통 흐름에 영향을 줄 수 있는 공항지상의 주요지점들을 의미하고, 링크는 노드와 노드를 연결하는 선을 의미한다[3].

Fig 1은 인천공항의 노드링크 모델로, 이를 개발하기 위해 우선 인천공항의 AIP를 바탕으로 유도도와 계류장 사이의 관제이양점(Transfer of Control Points, TCP) 및 관제 이양방향, 각 게이트별 push-back 절차, 활주로 운영 방향, 제방빙(de-icing) 시설 및 운영 절차 등 전반적인 지상

교통 운영 현황에 대한 사전 분석을 수행하였다 [4]. 그리고 사전 분석된 정보를 바탕으로 11개 노드 종류와 9개 링크 종류에 맞추어 공항 CAD 도면상에서 지상교통에 있어 주요한 지점에 노드 ID와 노드 종류를 부여하였으며, 이때 각 노드 ID는 중복되지 않도록 하였다. 또한 각 노드에 대해 CAD 좌표값을 추출하고 좌표변환을 통해 공항원점을 기준으로 하는 ENU(East-North-Up) 좌표값을 얻었다. 본 연구에서 노드 ID는 숫자로만 표기하였으므로 노드링크 모델을 기반으로 추출된 taxi route는 숫자로 표기된 노드 ID의 연속으로 표현되었다.

● 노드 종류

- GATE NODE: push-back이 필요한 게이트
- PARK NODE: 자력 출발이 가능한 게이트
- SPOT NODE: TCP 지점
- DEPARTURE NODE: 활주로 이륙 시작 지점
- ARRIVAL NODE: 활주로 도착 시작 지점
- RAMP NODE: 계류장내 항공기 이동 지점
- TAXI NODE: 유도로내 항공기 이동 지점
- QUEUE NODE: 출발 대기열 지역내 이동 지점
- RUNWAY CROSS NODE: 활주로 crossing 지점
- DEICING NODE: 제방빙지역 내의 pad 위치
- DEICINGTAXI NODE: 제방빙지역 내의 이동 지점

● 링크 종류

- GATE LINK: 두 GATE NODE를 연결한 링크
- RAMP LINK: 두 RAMP NODE를 연결한 링크
- TAXI LINK: 두 TAXI NODE를 연결한 링크
- RUNWAY CROSS LINK: 두 RUNWAY CROSS NODE를 연결한 링크
- QUEUE LINK: 두 QUEUE NODE를 연결한 링크
- DEPARTURE LINK: DEPARTURE NODE와 연결된 링크
- ARRIVAL LINK: ARRIVAL NODE와 연결된 링크
- SPOT LINK: SPOT NODE와 TAXI NODE를 연결한 링크
- DEICING LINK: 제방빙 지역 내의 NODE 간 연결 링크

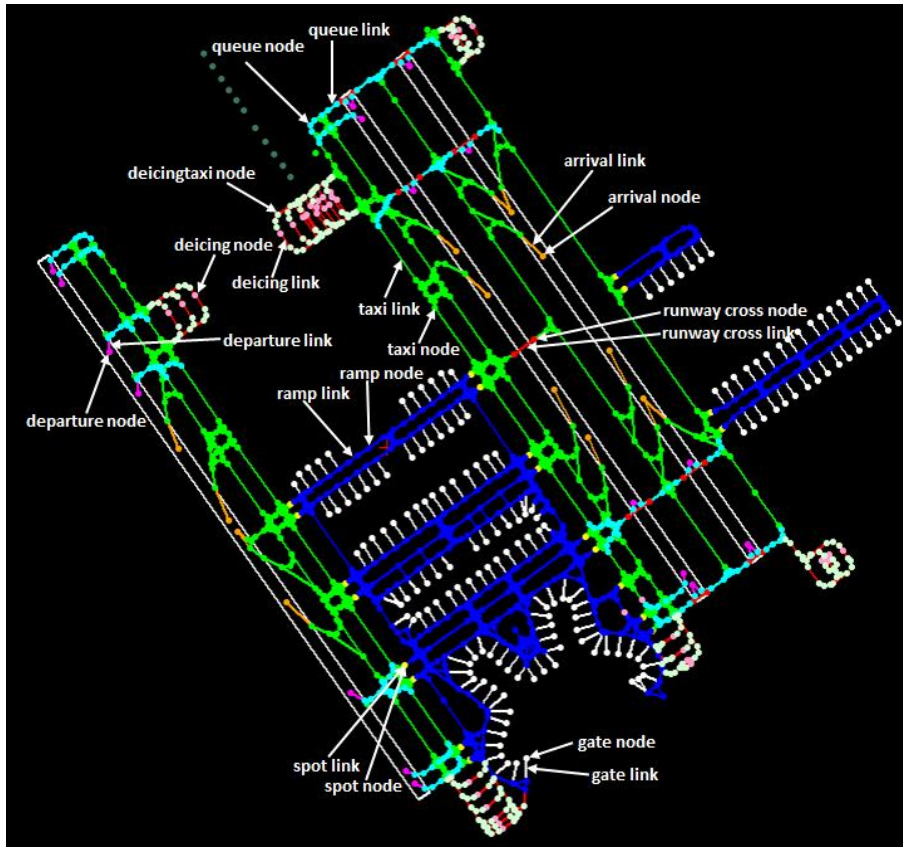


Fig 1. Node-link model of Incheon Airport

2. 경로 추출 방법

본 연구에서 항공기 지상이동 경로는 출발 항공기의 경우 항공기 출발 게이트(노드링크 모델의 GATE 또는 PARK NODE 위치)에서 활주로 이륙 지점(노드링크 모델의 DEPARTURE NODE 위치)까지, 도착 항공기의 경우에는 활주로 도착지점(노드링크 모델의 ARRIVAL NODE 위치)에서 게이트(노드링크 모델의 GATE or PARK NODE 위치)까지 항공기가 이동하면서 통과하게 되는 여러 노드 ID를 시간 경과 순서에 따른 시퀀스 형태로 구성하였다. 즉, 노드링크 모델상의 항공기 이동경로는 출발 항공기의 경우 GATE NODE 또는 PARK NODE가 source, DEPARTURE NODE가 sink가 되고, 도착항공기의 경우 ARRIVAL NODE가 source, GATE NODE 또는 PARK NODE가 sink가 되는 노드링크 네트워크 상의 이동경로이며, 노드 간 이동은 두 노드 사이에 사전 정의된 링크가

있는 경우에 해당 링크를 통해서만 가능하도록 생성되어야 한다.

본 장에서는 ASDE 항적 데이터를 노드링크 모델에 매치하여 자동으로 항공기 지상이동 경로를 추출하는 방법에 대해서 기술하였다. 특히 자동 매칭 과정에서, 실제 항공기의 이동경로를 나타내는 ASDE 항적과 노드링크 모델의 위치 오차로 인해 불필요한 노드들이 포함되거나 링크로 연결되지 않은 두 노드가 연속으로 추출되는 경우가 발생하는데, 이때 이렇게 부적절한 경로를 포함한 두 노드 사이의 경로를 적절히 수정하기 위해 Dijkstra 알고리즘을 적용하였다. ASDE 항적 데이터를 이용한 지상이동 경로 추출 전체 과정을 도식화 하면 Fig 2와 같으며, 이중 최단 경로 추출부분에 Dijkstra 알고리즘이 적용되었다.



Fig 2. Process flowchart for taxi route extraction

2.1 ASDE 항적 데이터 전처리

ASDE는 공항 지상을 감시하는 레이더 장비로 제공되는 항적 데이터에는 항공기 위치, 속도, 시간정보 등이 포함되어 있다. 그러나 통상적으로 ASDE 항적 데이터는 많은 잡음을 포함하고 있어 위치정보와 노드링크 모델을 바로 비교할 경우 큰 위치오차로 인해 매칭이 제대로 이루어지기 어렵다. 수집한 ASDE 항적 데이터는 off-line 데이터이므로 먼저 median filter를 적용하여 비정상적으로 큰 오차를 갖는 잡음 데이터를 제거하였다. 이렇게 1차적으로 잡음을 제거한 ASDE 항공기 위치정보와 노드링크 모델의 노드 좌표값을 비교할 경우 항공기 위치에 대응되는 노드를 쉽게 식별할 수 있다. 그러나 본론 제 1장에서 소개한 바와 같이 노드링크 모델에서는 공항 원점을 기준으로 한 ENU 좌표계로 각 노드의 좌표값을 설정한 데 반해, ASDE는 각 공항에 설치된 ASDE 장비에 따라 노드링크 모델의 좌표계와 다른 좌표계를 기준으로 항공기 위치정보를 제공할 수 있다. 따라서 이후 단계에서는 ASDE 항공기 위치정보를 좌표변환을 통해 노드링크 모델과 같이 공항 원점을 기

준으로 한 ENU 좌표계로 변환하는 작업을 수행하였다. 인천공항 ASDE의 경우 33L 활주로 시단을 원점으로 항공기 위치정보를 제공하고 있으며, 이들 정보를 좌표변환하여 노드링크 모델에 맵핑시킬 경우 Fig 3과 같은 결과를 얻을 수 있다. Fig 3에서 분홍색 점들은 ASDE에서 측정된 도착 항공기의 항적을 나타낸 것이다.

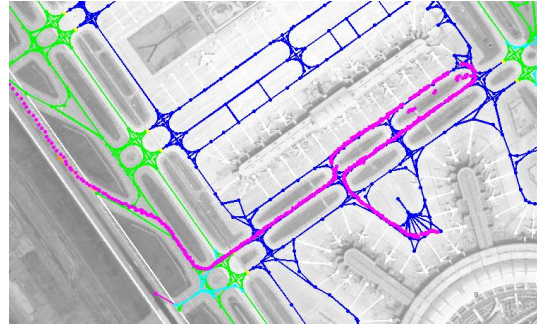


Fig 3. Node-link model overlaid with ASDE track

2.2 인접 노드 수집 및 중복 노드 제거

ASDE 항적을 노드링크 모델에 맵핑한 이후 “인접 노드 수집 및 중복 노드 제거” 단계에서는 ASDE 항적 데이터의 각 항공기 위치마다 항공기 위치 좌표값을 기준으로 반경 40m 이내에 위치한 노드를 노드링크 모델로부터 찾아 수집하는 과정을 수행한다. 이때 반경 40m는 인천공항의 유도로 및 계류장내 항공기 이동경로 간격을 고려하여 설정된 값으로 공항별 형상에 따라 달리 적용될 수 있다.

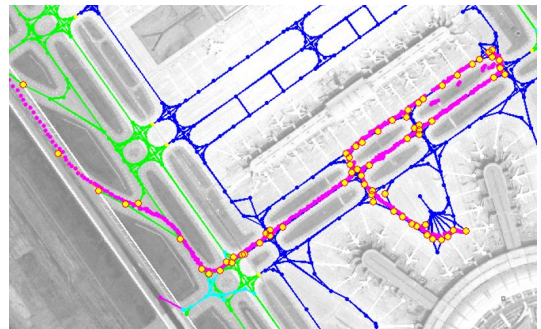


Fig 4. Node ID extraction adjacent to ASDE track

또한 인천공항의 ASDE는 1Hz로 항공기 위치정보를 제공하고 있기 때문에 항공기 위치 좌표

마다 반경 40m 이내에 위치한 노드를 수집할 경우 항공기 속도가 느린 계류장 및 유도로 지역에서는 동일한 노드가 중복되어 수집될 수 있다. 따라서 “인접 노드 수집 및 중복 노드 제거” 단계에서는 노드 수집과 함께 시간 경과에 따라 불필요하게 중복 수집된 노드를 제거하는 작업도 수행한다. Fig 3과 같이 도착 항공기 항적에 대해 이러한 과정을 수행하게 되면 최종적으로 Fig 4에 도시된 노란색 노드들을 얻을 수 있다. 즉 이 노드들은 ASDE 항적 데이터의 위치좌표와 40m 이내에 위치한 것으로 식별된 노드들로서, 노드 ID로 이루어진 taxi route를 추출하기 위한 초기 정보로 활용되게 된다.

2.3 경로의 시작 및 종료 노드 식별

본문 제2장 도입부에서 언급한 바와 같이 본 연구에서 taxi route는 출발 항공편의 경우 GATE 또는 PARK NODE에서 DEPARTURE NODE까지, 도착 항공편의 경우 ARRIVAL NODE에서 GATE 또는 PARK NODE까지로 그 영역을 한정하고 있다. 그러나 개별 항공편에 대한 실제 ASDE 항적 데이터는 때때로 여러 요인으로 인해 이들 영역에서 데이터 손실이 발생하는 경우가 종종 발생한다. 특히 게이트 주변 지역의 경우 건물들에 의한 신호간섭으로 인해 대부분의 항공편에 대해 항적정보가 거의 존재하지 않으며, 일부 항공편의 경우 활주로 부근에서도 항적정보가 손실되는 경우가 종종 발생하기도 한다. 따라서 항적정보만을 이용하여 taxi route 상의 노드들을 식별할 경우 항적이 존재하지 않는 구간에서의 노드를 식별하는데 어려움이 있다. 특히 taxi route의 시작 및 종료 노드의 경우 Dijkstra 알고리즘을 적용하기 위해 반드시 정확히 주어져야 하는 정보이기 때문에 “경로의 시작 및 종료 노드 식별” 단계에서는 항공편마다 배정된 실제 게이트 정보, 이륙 또는 착륙 활주로 정보를 활용하여 앞서 “인접 노드 수집 및 중복 노드 제거” 단계에서 얻어진 taxi route의 시작노드 및 종료 노드를 수정 및 보완하는 과정을 수행한다.

즉 “인접 노드 수집 및 중복 노드 제거” 단계에서 얻어진 taxi route내에 GATE NODE 또는 PARK NODE가 없을 경우 해당 항공편에 배정된 게이트 정보를 활용하여 해당 게이트에 매치되는 노드를 찾아 taxi route에 추가하고 GATE

NODE 또는 PARK NODE가 이미 존재하는 경우에는 실제 게이트 정보에 해당하는 노드 ID와의 일치 여부를 확인하여 그 값이 다를 경우 수정하는 과정을 거쳐게 된다. 마찬가지로 방법으로 taxi route 내에 DEPARTURE NODE 또는 ARRIVAL NODE가 없는 경우에는 해당 항공편에 배정된 이륙 활주로 또는 착륙 활주로 정보를 활용하여 이에 매치되는 노드를 찾아 taxi route를 보완해 주고, DEPARTURE NODE 또는 ARRIVAL NODE가 taxi route 내에 이미 존재하는 경우 배정 활주로 정보와의 일치 여부를 확인하여 수정/보완한다. Fig 5에서는 앞서 Fig 4에서 얻어진 taxi route 정보에 해당 항공편에 배정된 게이트 및 활주로 정보를 활용하여 경로상의 첫 번째 노드인 ARRIVAL NODE가 식별되고, 경로상의 마지막 노드인 GATE NODE가 추가되었다. 이들 노드들은 Fig 5에서 빨간색 점으로 나타내었다.

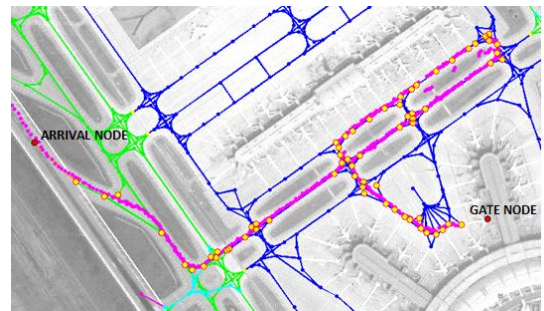


Fig 5. Start and last node of taxi route

2.4 Invalid 노드 제거

앞서 노드링크 모델 개발 시 각 노드 ID는 서로 중복되지 않도록 부여하였기 때문에 두 노드 간을 연결한 링크들 또한 노드링크 모델 내에서 중복되지 않도록 정의되었다. 이러한 가운데 “Invalid 노드 제거” 단계에서는 “경로 시작 및 종료 노드 식별” 단계를 거쳐 얻어진 taxi route를 기준으로 앞뒤 노드를 서로 연결한 링크를 생성하였을 때 해당 링크들이 노드링크 모델에서 정의한 링크인지 여부를 확인하여 정의되지 않은 링크일 경우 taxi route 내에서 이들 노드들을 제거하도록 하였다. 또한 유효하지 않은 노드들을 제거한 이후 taxi route 내에 남은 노드들을 다시 연결하여 노드링크 모델에서 정의한 링크 정보와 비교하고, 두 노드 Node1,

Node2 사이에 누락된 노드들이 노드링크 모델에 존재하는지 여부를 확인한 후 Node1, Node2 두 노드 간 적절한 상세 경로를 추가할 수 있도록 하였다. 즉 이 단계에서는 노드링크 모델에서 정의한 링크정보를 바탕으로 taxi route상에 포함된 불필요한 노드는 제거하고 누락된 노드는 추가하는 작업을 수행한다. Fig 6은 “경로 시작 및 종료 노드 식별” 단계를 통해 식별된 taxi route를 바탕으로 “Invalid 노드 제거” 단계를 거치고 남은 노드들을 도시한 것으로, Fig 5와 비교하여 Fig 6 상의 A, B 구역에서 몇몇 노드들이 제거된 것을 볼 수 있다.

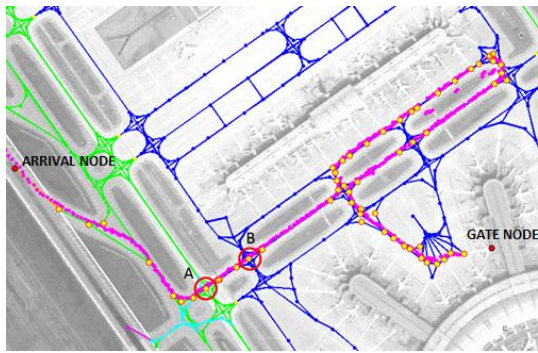


Fig 6. Removing the invalid node contained in the taxi route

2.5 경로 세부 구간 나누기

Fig 6을 보면 앞의 여러 과정을 통해 추출된 taxi route 상에 여전히 불필요한 노드들이 많이 포함되어 있는 것을 볼 수 있다. 따라서 이러한 불필요한 노드를 제거하기 위해, 최단경로 문제에 많이 이용되고 있는 Dijkstra 알고리즘을 적용하였다. Dijkstra 알고리즘은 가중치가 있는 방향 그래프에서 임의의 두 노드사이의 최단거리를 구하는 알고리즘으로, 음수 가중치가 없는 경우에만 적용될 수 있다[5]. 본 연구의 taxi route 모델링에서도 노드와 노드간의 거리(즉, 링크 길이)를 가중치로 적용하고 Dijkstra 알고리즘을 적용하여, 앞서 얻은 taxi route내 시작 노드에서부터 종료 노드에 도달하기까지 항공기 이동이 가능한 모든 경로를 탐색하여 그 중 링크 길이의 합이 가장 적은 경로를 식별해내도록 할 수 있다.

단, Dijkstra 알고리즘을 앞서 얻어진 taxi

route내 시작 노드에서 종료 노드 전체 구간에 대해 한 번에 최단경로를 탐색하도록 적용할 경우 시작 노드와 종료 노드에 따라 이동거리가 최단인 단일 경로만이 생성된다. 본 연구에서는 ASDE 항적 데이터를 이용하여 Fig 6과 같이 최단 경로가 아니더라도 실제 상황에서 지상교통 혼잡도에 따라 변형되는 다양한 taxi route를 모델링 하고자 하는 목적이 있으므로, taxi route 전체 구간에 대해 한번에 Dijkstra 알고리즘을 적용하는 것은 적절치 않다고 볼 수 있다. 따라서 “경로 세부 구간 나누기” 단계에서는, 공항 형상 정보를 이용하여 “Invalid 노드 제거” 단계에서 얻은 taxi route를 여러 세부 구간으로 나누어, “최단경로 추출” 단계에서 이들 각 구간에 대해 Dijkstra 알고리즘을 적용함으로써, 불필요한 노드들이 경로에 삽입되거나 경로 상에 링크로 연결되지 않은 노드가 연속되어 나타나는 경우 등, 노드링크 모델에 적절한 경로가 아닌 세부 구간들이 다시 한 번 적절하게 수정될 수 있도록 하였다.

경로 세부 구간을 나누는 방법은 각 공항의 형상에 따라 달리 적용될 수 있는데, 인천공항의 경우 활주로와 유도로가 서로 평행하고, 계류장내 주요 항공기 지상이동 경로가 활주로 및 유도로에 수직인 점을 고려하여 “Invalid 노드 제거” 단계에서 수집된 노드를 이용하여 앞/뒤 노드 간 항공기 이동에 따른 방위각(heading) 변화를 우선 계산하고, 계산된 방위각 변화량 값이 3회 이상 변하지 않을 경우(즉 일정 구간 항공기가 직선 이동한 경우) 방위각 변화가 발생하지 않은 구간의 시작 및 종료 노드를 식별하도록 하였다. 따라서 Fig 6의 “Invalid 노드 제거” 단계를 통해 얻은 taxi route에 대해 이러한 과정을 거쳐 Fig 7과 같이 직선 이동 구간을 찾고, 직선 이동 구간의 시작 및 종료 노드를 식별하여 검정색으로 나타내었다.

다음 “경로 세부 구간 나누기” 단계에서는 Fig 7에서 식별된 검정색 노드들을 이용하여 Fig 8과 같이 박스 형태로 경로 세부 구간을 나누게 되는데, 이때 이들 박스는 빨간색 및 검정색 박스와 같이 각 구간의 시작 및 종료 노드를 포함하는 최소한의 크기로 생성되도록 한다. 이러한 과정을 거칠 경우 앞의 “Invalid 노드 제거” 단계를 통해 얻은 taxi route는 총 7개의 세부 경로 구간으로 나누어지게 된다.

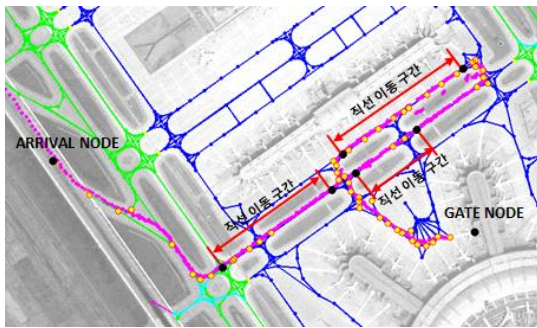


Fig 7. Straight segment identification of taxi route



Fig 8. Taxi route segments

2.6 최단경로 추출

“최단경로 추출” 단계에서는 앞의 “경로 세부 구간 나누기” 단계에서 식별된 경로 각 세부 구간 즉 Fig 8에 표기된 각 박스에 대해 Dijkstra 알고리즘을 적용하여 최단경로를 추출하는 단계이다. Dijkstra 알고리즘은 Fig 9와 같이 노드와 링크로 이루어진 네트워크가 존재한다고 하였을 때 A가 경로의 시작점이고 F가 경로의 종료점이며, 화살표 진행방향으로만 노드 간 이동이 가능하고 두 노드 사이의 거리는 링크 길이와 같다고 할 경우, 노드 A에서 노드 F로 이동 가능한 다양한 경로에 대해 총 경로 이동거리를 계산하여 그 중 경로 이동거리가 가장 짧은 경로를 찾아내는 알고리즘이다[5]. Fig 9의 경우 빨간색 링크로 이루어진 경로(A→D→C→F)가 노드 A→F까지의 최단이동경로가 된다.

이러한 Dijkstra 알고리즘을 “경로 세부 구간 나누기” 단계에 적용하기 위해서는 우선 Fig 8의 각 세부 구간에 대해 해당 박스 영역 내에 위치한 노드링크 모델상의 여러 노드와 링크를

식별하고, 이들을 이용하여 Fig 10의 빨간색 노드와 노란색 링크로 이루어진 네트워크를 생성한다. 이때 네트워크 생성을 위해 앞서 식별된 Fig 6의 taxi route 노드 대신 경로 세부 구간의 해당 박스 영역 내에 위치한 노드링크 모델상의 여러 노드를 사용하도록 한 것은 ASDE 항적 데이터가 해당 세부 구간 내에서 일정 부분 존재하지 않더라도 그 구간에 대한 최단이동경로를 추출하기 위해서이다. 실제 인천공항에서 수집된 ASDE 항적 데이터의 경우 신호간섭이 자주 발생하는 계류장 내에서 짧은 시간동안 잦은 항적 데이터 손실이 발생하고 있으며, 이러한 항적 데이터로부터도 정상적인 taxi route를 추출하기 위해 해당 방법을 적용하였다. 이 방법의 경우 네트워크 내 노드 수 증가로 인해 약간의 연산속도 저하가 발생하지만, 보다 많은 taxi route를 추출할 수 있다는 장점을 가지고 있다.

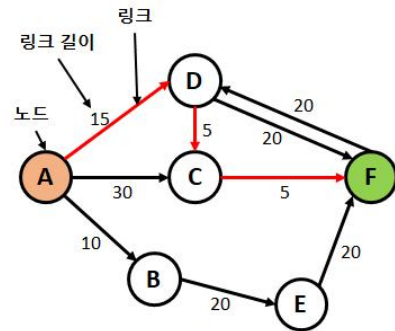


Fig 9. Network components of the Dijkstra algorithm

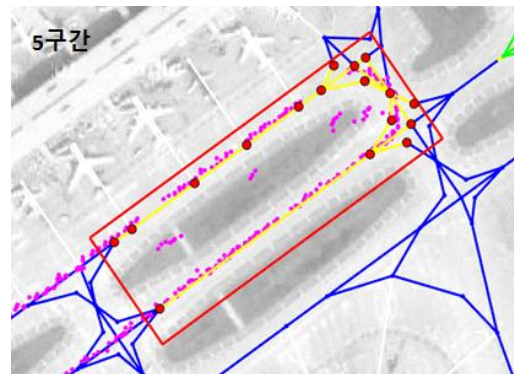


Fig 10. Creating network for each segment of taxi route

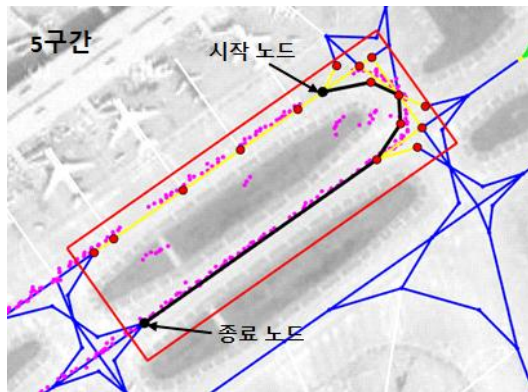


Fig 11. Shortest path extraction

이렇게 생성된 네트워크에 대해 Dijkstra 알고리즘을 적용하면 Fig 11의 검정색 선으로 표시된 최단이동경로를 찾을 수 있다. 이때 경로 시작 및 종료 노드는 앞의 “경로 세부 구간 나누기” 단계에서 식별된 각 경로 세부 구간의 시작 및 종료 노드를 이용하게 되며, 이는 Fig 11에 검정색 점으로 나타내었다. 이러한 과정을 Fig 8의 7개 구간 전체에 걸쳐 수행한 후 얻어진 각 세부 구간 최단이동경로를 서로 연결할 경우 Fig 12와 같이 게이트부터 활주로까지 이어지는 완전한 taxi route를 얻을 수 있다.

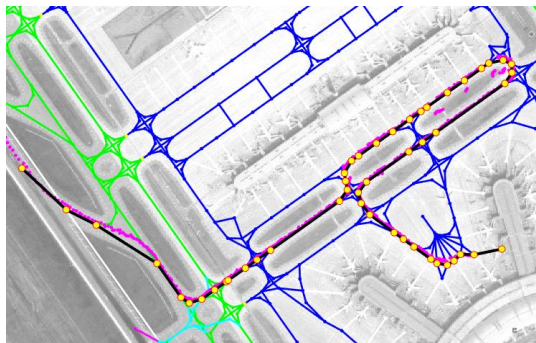


Fig 12. Final extracted taxi route

3. 지상이동 경로 추출 결과 분석

2장에서 소개한 방법을 이용하여 인천공항에서 수집된 2015년 4월의 ASDE 항적 데이터에 대해 노드링크 모델을 기반으로 항공기 지상이동 경로를 추출하였으며, 그 결과를 바탕으로 지상이동 경로 수, 동일 지점 간 지상이동 경로 종류의 수, 유도도 경로 사용 빈도 등을 분석하

였다.

3.1 지상이동 경로 수

인천공항의 경우 2015년 4월 한 달간 25,461대(출/도착 포함 1일 평균 849대) 항공편에 대해 ASDE 항적데이터가 수집되었으며, 그 중 92.6%에 해당하는 23,579 항공편에 대해서만 지상이동 경로가 추출되었다. 지상이동 경로가 추출되지 못한 약 7%의 ASDE 항적데이터의 경우 대부분 항적 데이터 자체가 중간에 장시간 손실된 부분이 많아 정상적인 경로 추출이 불가능하였다. 23,579개의 지상이동 경로 가운데 중복되는 경로를 제외할 경우 약 3,386개의 서로 다른 지상이동 경로를 얻을 수 있었으며, 이중 출발 항공편에 대한 경로 수는 1,910개, 도착 항공편에 대한 경로 수는 약 1,476개로 나타났다.

3.2 동일 지점 간 지상이동 경로 종류의 수

출발 항공편의 경우 특정 게이트에서 특정 활주로 이륙지점까지, 도착 항공편의 경우 특정 활주로 착륙지점에서 게이트까지 항공기가 지상이동을 함에 있어 다양한 지상이동 경로를 가질 수 있는데, 이와 같은 경로들에 대한 분석을 수행하였다. 앞서 인천공항의 경우 출발 항공편에 대해 약 1,910개의 지상이동 경로 수를 가지고 있다고 하였는데, 그 중 동일 게이트에서 동일 활주로 이륙지점으로 이동하는 경로들의 수에 대해 분석하였으며 그 결과는 Fig 13과 같다. Fig 13의 X축은 동일 지점 간을 이동하기 위한 항공기 지상이동경로의 수를 의미하고, Y축은 그 빈도 수를 의미한다. 즉, Fig 13의 분석결과를 보면, 인천공항의 경우, 324개의 게이트-활주로 조합에 대해서는 오직 한 개의 지상이동 경로만이 존재하였으나, 일부 게이트-활주로 조합(2가지)에 대해서는 추출된 지상이동경로의 수가 최대 12개까지 발생하기도 하였음을 확인할 수 있다. 도착 항공편에 대해서도 동일한 분석을 수행하였으며, 그 결과 동일 활주로 착륙지점에서 게이트까지 최대 8개의 다양한 지상이동 경로가 존재함을 확인하였다. 출발의 경우 push-back 방향에 따라서도 지상이동 경로가 조금씩 변경될 수 있기 때문에 도착에 비해 최대 지상이동경로가 좀 더 다양하게 나타난 것으로 추측된다.

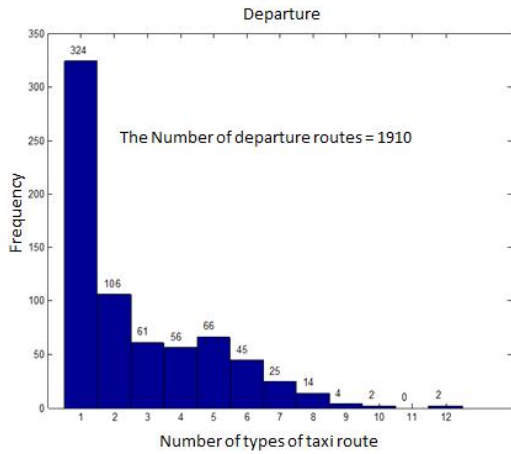


Fig 13. Number of taxi routes from a specific gate to a specific runway

3.3 유도로 경로 사용 빈도

마지막으로 유도로 경로 사용 빈도에 대한 분석을 수행하였는데, 본 연구에서 유도로 경로라 함은 노드링크 모델의 SPOT → DEPARTURE 노드 또는 ARRIVAL → SPOT 노드로의 이동 경로를 의미한다.



Fig 14. Departure route on taxiway



Fig 15. Arrival route on taxiway

분석 결과, 인천공항 출발 항공편의 경우 중복 유도로 경로를 제외할 경우 총 193개의 유도로 이동 경로가 존재함을 확인하였고, 도착 항공편의 경우 총 106개의 경로를 확인할 수 있었다. Fig 14와 Fig 15에 나타난 파란색 선은 인천공항 출발 및 도착 항공편 각각에 대한 유도도 이동 경로 중 가장 많이 사용된 경로 20개를 구글 지도상에 나타낸 것이다. 이와 같은 분석들을 통해 인천공항 유도도 중 어느 유도도가 항공기들에 의해 더 많이 사용되는지를 쉽게 식별할 수도 있다.

III. 결 론

본 논문에서는 노드링크 모델을 기반으로 인천공항의 ASDE 항적 데이터와 Dijkstra 알고리즘을 이용하여 자동으로 지상이동 경로를 추출하는 방법에 대해 기술하였다. 추출된 지상이동 경로는 실제 인천공항에서 항공기 지상이동시 사용되고 있는 경로로서, DMAN의 핵심 기능 중 하나인 항공기 지상이동시간 예측 시 보다 실제에 가까운 예측이 가능하도록 할 수 있으며, 개발 중인 DMAN의 확장된 기능의 일부인 지상이동 경로 스케줄링에 활용될 수 있다. 또한 추출된 인천공항의 다양한 지상이동 경로를 이용하여 지상이동 경로 수, 동일 지점 간 지상이동 경로 종류의 수, 유도로 경로 사용 빈도

등을 분석함으로써, 인천공항의 항공기 지상이동 운영 현황을 살펴 볼 수 있었다. 본 연구에서 추출된 지상이동 경로는 상기 기술한 목적 이외에도 공항지상교통 관제시뮬레이터 개발 등에 활용될 수 있을 것이다.

후 기

본 연구는 국토교통부 연구과제인 '항공기 출발 및 도착 통합 관리 기술 연구'와 미래창조과학부 주요사업 연구과제인 '유무인기 복합항행 기반기술 연구'과제에 의해 연구비 지원을 받았으며, 이에 감사드립니다.

Reference

- [1] ICAO, Working Document for the Aviation System Block Upgrades - The Framework for Harmonization, March 2013.
- [2] Jeon, D., Kim, H., Eun, Y., and Oh, E., "Development of Integrated Departure/Arrival Management Technologies - R&D Planning Document", Korea Aerospace Research Institute, October 2014.
- [3] Jeong, M., Lee, J., Eun, Y., and Jeon, D., "Node-Link Model Generation of Incheon International Airport" Proceeding of the 2015 KSAS Fall Conference, 2015.
- [4] Ministry of Land, Infrastructure, and Transport of Korea, "AIP Republic of Korea", RKSI AD 2, May 2014.
- [5] Cormen, T., Leiserson, C., Rivest, R., and Stein, C., "Introduction to Algorithms", 3rd Edition, 2009.