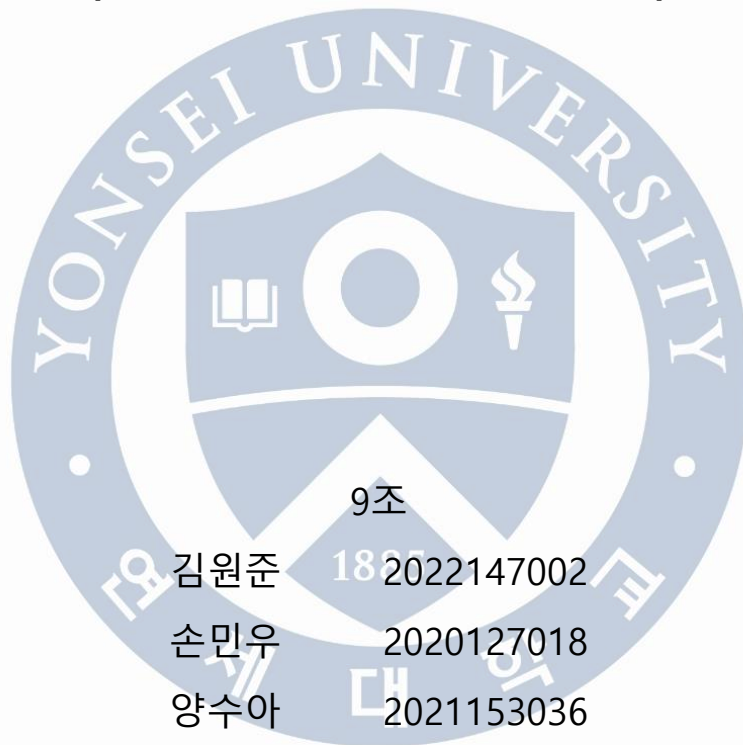


2023-2 시뮬레이션 기말 프로젝트
공항에서 활용되는
최소 환승 시간(Minimum Connection Time)의 유의성 검증



9조

김원준	2022147002
손민우	2020127018
양수아	2021153036
한준희	2021171001

김원준 SIN part; Conceptualization, Formal Analysis, Data Curation, Investigation, Methodology, Software, Validation, Writing–Original Draft, Writing-Review&Editing; 손민우 DFW part; Formal Analysis, Data Curation, Investigation, Methodology, Software, Validation, Writing–Original Draft, Writing-Review&Editing; 양수아 ICN part; Formal Analysis, Data Curation, Investigation, Methodology, Software, Validation, Writing–Original Draft, Writing-Review&Editing; 한준희 LAX part; Formal Analysis, Data Curation, Investigation, Methodology, Software, Validation, Writing–Original Draft, Writing-Review&Editing

목차

1. 개요

- A. 주제 선정 배경
- B. 프로젝트의 중요성

2. 목표

- A. 현재 시스템
- B. 목표 정의
- C. 상세 사항

3. 시뮬레이션 모델

- A. 개요
 - i. 대상
 - ii. 모델 검증
 - iii. 파일 설명
- B. 로스엔젤레스 국제 공항(LAX)
 - i. 공항 설명
 - 1. 공항 구조
 - 2. 특이사항
 - ii. 시뮬레이션 모델 설명
 - 1. 전체 구조
 - 2. 구조 가정
 - 3. 데이터 가정
 - 4. 모델 세부 설명
 - 5. 모델 검증 계획
- C. 달라스 포트워스 국제 공항(DFW)
- D. 창이 국제 공항 (SIN)
- E. 인천 국제 공항 (ICN)

4. 수치적 실험

- A. 실험 계획
- B. 실험 결과
 - i. 로스엔젤레스 국제 공항(LAX)
 - ii. 달라스 포트워스 국제 공항(DFW)
 - iii. 창이 국제 공항 (SIN)
 - iv. 인천 국제 공항 (ICN)

5. 결론

- A. 결과 요약 및 분석
 - i. MCT의 적정 여부 판단 기준 - 가설 검정을 바탕으로
 - ii. 구체적인 공항 모델별 가설 검정
- B. 의의 및 한계
 - i. 의의
 - ii. 한계
- C. 현실에의 적용
- D. 프로젝트 심화 제안

1. 개요

A. 주제 선정 배경

- 항공 교통에서 활용되는 최소 환승 시간이 실제로 의의를 가지는 것인지 검증이 필요하다.
- 오늘날 항공 교통에서 환승은 중요한 부분을 차지하고 있고 많은 이용객들이 환승을 경험한다. 이와 같은 배경에서 중심으로 고려되어야 할 부분은 '환승에 필요한 최소한의 시간'일 것이다. 이는 공항에 도착한 후 환승에 성공하기 위해 필요한 최소한의 시간이다. 관련해서 국제 항공 운송 협회(IATA, International Air Transport Association)는 "승객과 위탁수화물이 최초 탑승편에서 연결편으로 탑승 및 이동하는데 필요한 최소한의 시간"으로 '최소 환승 시간(Minimum Connection Time, 이하 'MCT')' 개념을 말하고 있다.
- 이는 곧 항공기 탑승객이 공항에서 다른 항공기로 환승할 수 있음을 보장하는 최소한의 시간을 설정한 것이며, 다른 말로 이해하자면 일반적으로 환승에 성공하기 위해서는 탑승편의 도착과 연결편의 출발 사이에 최소한 MCT의 시간이 필요하다는 의미이다. 따라서 이용객들은 환승 항공편 예약에서 MCT를 고려하기도 하며, 시스템 상으로도 MCT 보다 그 사이의 여유 시간이 짧은 환승 항공편은 예약이 불가능하도록 제한을 두고 있기도 하다.
- 결국, 개인 및 기업에게 있어 항공편의 환승 가능 여부를 판단하는데 공항의 MCT는 주요한 요소로 활용되고 있는 것이다. 이와 같은 배경에서 과연 MCT가 실제로 유효한 기능을 발휘하고 있는지 의심해 볼만 하다. 달리 말하여, 일반적으로 이용객들이 MCT에 해당하는 시간 내에 환승에 성공하는지 검토하는 과정은 유의미할 것이다.

B. 프로젝트의 중요성

i. 환승 가능 여부 판단 기준으로 MCT 활용

- MCT는 환승 가능 여부를 판단하고 그에 따라 비행편을 예약 및 구입하는 과정에서 활용되기에, 이것이 정확한지 확인할 필요가 있다.
- 달리 서술하여, MCT를 기준으로 항공편 예약이라는 비용 지출을 결정하게 되는 만큼, MCT에 대한 보다 엄밀한 검토가 요구된다.
- 애당초 문제를 내재하고 있는 MCT를 근거로 비용을 지출하여 손해를 보는 경우가 발생해서는 안되기 때문이다.

ii. MCT의 신뢰성 문제 대두

- 현실에서 MCT를 충족했음에도 환승을 하지 못하는 사례가 발생한다.
- 위에서 언급하였듯, 기존에 통용되고 있는 MCT는 이용객 및 기업이 최초 탑승편과 연결편 간의 환승 가능 여부를 판단하는 것에 활용되고 있다. 전자의 도착과 후자의 출발 사이의 시간이 MCT보다 길 경우 환승이 가능한 것으로 고려된다.

- 그러나 실제 사례에서는 그와 같이 환승이 가능한 것으로 판단되더라도 환승에 실패하기도 한다. 연결시간이 MCT보다 길어 환승이 가능할 것으로 보고 해당 일정을 계획하였으나, 막상 실제 환승을 경험하니 곧장 환승 게이트로 향하더라도 환승을 하지 못하는 경우이다.

- 그의 한가지 예로, 한 여행사에서 MCT를 고려해 항공편을 예약하였으나, 실제 승객은 베이징 공항에서 MCT 보다 긴 연결시간이 있었음에도 환승에 실패하였다.¹

- 이외에도, 이용객들은 환승 가능 여부를 판단할 때에 MCT를 고려하기는 하지만 이에 전적으로 의존하지는 않는 경향이 존재한다. 타지에서 겪는 환경 변화로 인한 어려움을 포함한 기타 돌발상황으로 인한 이동 시간 증가를 MCT가 포함하지 못하는 것이 그 이유이다. 이용객들은 MCT를 고려하기는 하지만 온전히 신뢰하지 못하고 이에 2-3시간 정도를 더하여 연결시간을 계획한다.

- 이처럼 기존 MCT에서 나타나는 신뢰성 문제는 더 나아가 이의 유효성에 대한 의문으로 이어지는 실정이다. 이러한 맥락에서 기존의 MCT가 진정한 가치를 지니고 있는지에 대해 본 프로젝트를 통해 검토해볼 필요가 있다.

iii. 승객 및 항공사와 직접적 관련

- 현재 신뢰성이 우려되는 MCT는 더 나아가 환승객 개인과 더불어 항공사 등의 기업에게 피해를 초래한다.

- MCT 기준을 반영하여 계획된 환승 일정에서 환승에 실패하게 되면 경우에 따라서 환승객 개인은 다음 항공편을 위한 지출을 부담할 수 있으며, 항공사 측은 이에 대한 보상을 해야 할 의무가 생기기도 한다. 최악의 경우에는 책임이 누구에게 있는지를 판별하게 위해 법적 분쟁으로도 이어진다.

- 이러한 상황에서 그 근본적 원인은 당초에 환승이 가능한 것으로 판단하게끔 한 MCT에 있다고 할 것이다. 결국, 환승객과 항공사 등 기업이 입는 피해가 정확하지 못한 MCT에 있는 것이다. 따라서 이와 같은 MCT를 검토해볼 필요가 더욱이 존재한다고 말할 수 있다.

¹ 진정미, "베이징 공항 '환승 65분'...환승시간 모자라 탑승 못해", 컨슈머치, 2023/11/26, <https://www.consumuch.com/news/articleView.html?idxno=61198>.

2. 목표

A. 현재 시스템

- i. MCT는 공항 별로 상이하며, 구체적인 산정 기준이 정확하게 알려진 바가 없다. 또한 MCT는 기본적인 수속, 터미널 간 이동 프로세스, 공항의 구조 등을 고려하여 자체적으로 협의하여 설정된다 알려져 있다.²
- ii. 본 프로젝트에서는, MCT를 설정하는 과정에 있어서 환승 중에 발생할 수 있는 여러 돌발상황 (수하물 수령에 긴 시간이 소요되는 경우, 입국/보안 검사 등의 과정에서 발생할 수 있는 추가적인 지연 시간 등)을 고려하지 않았을 가능성이 있다고 판단하였다. 즉, 기존에 설정된 MCT는 단순히 도착 게이트에서 출발 게이트까지의 이동 시간과 이상적인 소요시간을 중점으로 하여 고려하는 것으로 보인다.
- iii. 즉, MCT가 실제 현장 상황을 구체적 반영하기 보다는 앞서 말했듯 환승 과정에서 문제가 없는 이상적인 상황을 가정하고 설정되었을 가능성이 존재하는 것이다.³
- iv. 하지만 이렇게 선정된 MCT를 믿고 따르는 소비자들 중 환승 시간을 지키지 못해 경유지에서 제 때 환승하지 못하는 경우가 발생한다. 이로 인해 이후의 항공편을 이용해 환승해야 하는 상황에 놓이기도 하며, 실제로 항공사들이 이를 대비해 예비 항공편을 준비해두는 경우도 있다.⁴

B. 목표 정의

- i. **공항을 선별하여, 실제 공항 내에서의 MCT가 유효한지 판단할 것이다.** 만약 MCT가 유효하다면 대부분의 환승객들이 그 시간안에 환승에 성공해야 할 것이다. 따라서 그러한 MCT의 유의성 검증 과정에서 실제로 환승객들이 어느 정도의 비율로 MCT 시간 내에 환승을 성공적으로 수행하는지 확인해보고자 한다.
- ii. MCT가 설정된 근거는 공개된 자료가 없어 판단하기 어렵다. 그렇기 때문에 실제로 설정된 MCT가 시뮬레이션으로 파악할 수 있는 공항의 상황을 반영한 것인지, 임의적으로 정한 수치에 불과한 것인지 간접적으로 검증해보고자 한다.
- iii. MCT가 공항의 실제 상황을 반영하지 못하고 임의적으로 설정된 것으로 확인된다면, MCT가 실제로 공항의 환승객들에 대해서 유의미한 가이드라인을 제공

² [RCED-88-207 Airline Scheduling: Airline Practices in Establishing and Maintaining Connecting Times \(gao.gov\)](#)

³ [What is the MCT \(Minimum Connection Time\) to avoid missing a connection? \(travelguys.fr\)](#)

⁴ [RCED-88-207 Airline Scheduling: Airline Practices in Establishing and Maintaining Connecting Times \(gao.gov\)](#)

할 수 있는지에 관해 검증해보고자 하였다.

C. 상세 사항(시뮬레이션의 필요성)

- i. 각 공항은 혼잡도와 구성이 다르고, 특정 상황에서 변칙적인 대기열이 발생하는 등의 예측할 수 없는 문제가 존재하기에 직접 시뮬레이션 해봄으로써 실제 검증이 필요하다. 따라서 시뮬레이션을 통해 단순히 수치적 계산으로는 추정할 수 없는 실제 현장의 불확실성을 고려하여 MCT를 검증해보고자 했다.

3. 시뮬레이션 모델

A. 개요

i. 대상

- 공항 간 유의미한 대조를 위하여 환승 시간이 오래 걸릴 것으로 예상되는 공항 2곳과, 비교적 짧은 환승 시간이 필요할 것으로 판단되는 공항 2곳을 선정하였다.
- 전자로, 출입국 심사가 까다롭고 환승 과정에서 출입국 심사를 받아야 하는 미국 공항 중, 이용객이 많은 댈러스-포트워스 국제공항과 로스앤젤레스 국제공항을 대상으로 선정하였다.
- 후자로는, 아시아권에서 허브 공항으로 기능하며 간편한 환승체계를 구축한 것으로 알려진 싱가포르 창이 공항과, 우리나라의 인천 국제 공항을 선정하였다.

ii. 모델 검증(verification/validation) 계획

- 각 공항의 모델 별로 상황에 맞게 4장에서 검증을 진행한다. 각 모델 별 검증 계획은 아래의 각 모델에 대한 서술에서 기술하겠다.

iii. 파일 설명

- 각 모델과 관련된 파일은 아래의 표와 같다.

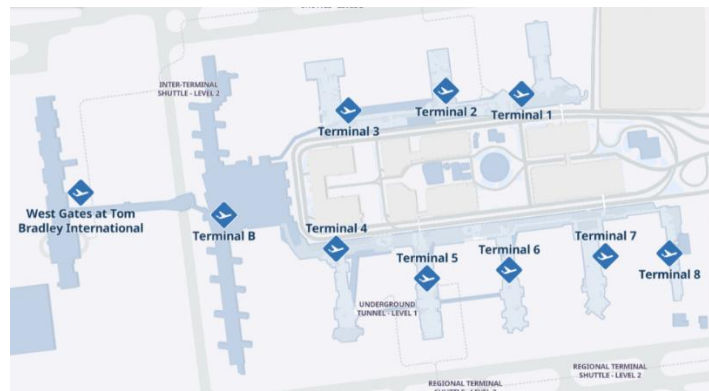
	LAX	DFW
모델(doe)	LAX_model.doe	DFW_model.doe
시뮬레이션 결과 리포트(pdf)	LAX_report_overview.pdf	LAX_report_overview.pdf
아웃풋 데이터 (csv)(Readwrite 모듈을 통해 도출된 파일)	LAX_output.csv	DFW_output.csv, DFW_tsa_quetime.csv
	SIN	ICN
모델(doe)	SIN_model.doe	ICN_model.doe

시뮬레이션 결과 리포트(pdf)	SIN_report_entities.pdf, SIN_report_queue.pdf, SIN_report_user_specified.pdf	ICN_report_overview.pdf
아웃풋 데이터 (csv)(Readwrite 모듈을 통해 도출된 파일)	SIN_output.csv	ICN_output.csv

B. 로스앤젤레스 국제공항(Los Angeles International Airport, LAX)

i. 공항설명

1. 공항구조



- 로스앤젤레스 국제공항(이하 'LAX')는 위의 그림과 같이 터미널1(이하 'T1'), 터미널2(이하 'T2'), 터미널3(이하 'T3'), 터미널4(이하 'T4'), 터미널5(이하 'T5'), 터미널6(이하 'T6'), 터미널7(이하 'T7'), 터미널8(이하 'T8'), 터미널B(West Gates at Tom Bradley International 포함, 이하 'TBIT'), 총 9개의 터미널로 구성되어 있다.

- TBIT는 대부분의 항공편을 국제선으로 운용하고 있으며, 그 외의 터미널은 이와 반대로 대부분 국내선 항공편을 운용하고 있다.

- TBIT에는 현재 총 33개의 게이트를 운영 중이다.

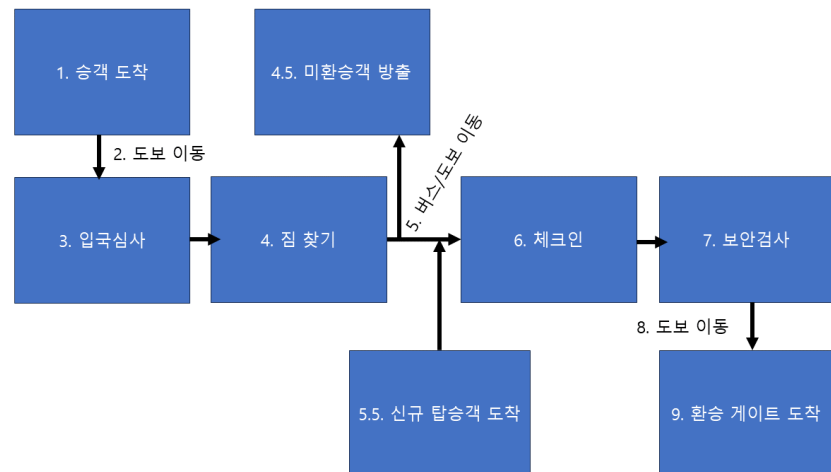
2. 특이사항

- 터미널들이 둘러싼 중앙 부분은 중심으로 터미널 간 운행하는 버스가 존재한다. 이는 TBIT->T4방향의 단방향으로 순환 운행한다.

- TBIT의 경우, 짐 찾는 곳 및 입국장이 한 곳으로 모여 있어, TBIT를 통해 도착하는 모든 승객들은 해당 위치로 모이게 된다.

ii. 모델설명

1. 전체 구조



- 모델의 전체 구조는 위와 같다. [1] 각 게이트를 통해 승객(환승객, 미환승객 포함)이 도착한다. [2] 짐 찾는 곳까지 도보로 이동하여, [3] 짐을 찾고 [4] 입국심사를 받는다. 이후 [4.5] 환승하지 않는 승객은 방출되고 [5] 환승하는 승객은 각자에게 할당된 터미널로 이동한다. [5.5] 각 터미널에서는 신규 탑승객들이 추가되며, [6] 환승객은 이들과 섞여 체크인과 [7] 보안검사를 진행한다. 이들이 완료된 후, [8] 도보 이동을 거쳐 [9] 최종적으로 환승할 게이트에 도착한다.

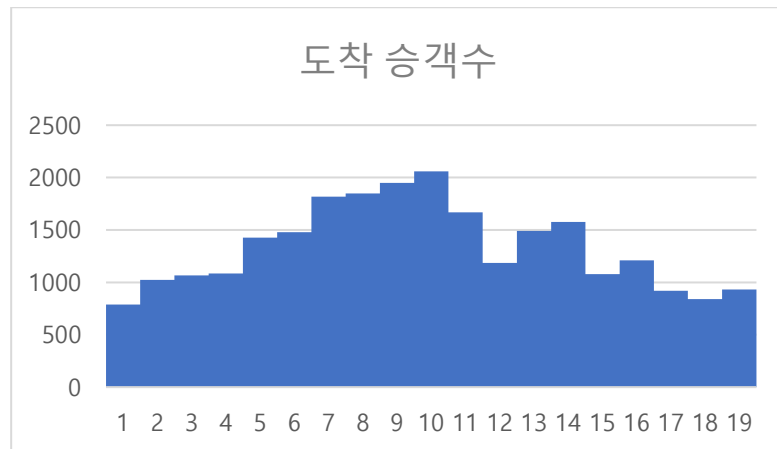
2. 구조 가정

- 국제선-국내선 환승 경우만 존재한다.(국제선-국제선, 국내선-국제선, 국내선-국내선의 경우도 존재하나, 최우선적으로 문제가 되는 경우는 가장 절차가 복잡한 국제선-국내선의 경우이므로, 해당 경우만 고려한다.)
- TBIT에서는 오직 국제선 항공편만 운영되고 그 외 터미널에서는 오직 국내선 항공편만 운영된다. (LAX 공식 사이트를 참고하면, TBIT에서도 국내선 항공편이 존재하고 그 외 터미널에서도 국제선 항공편이 존재하지만, 각각의 터미널에서 국제선과 국내선의 비율이 압도적이기에 그와 같이 가정한다.⁵⁾)
- 국제선 도착 항공편은 TBIT의 모든 게이트에 걸쳐 고르게 도착한다.
- T1, T2, T3, T4, T5, T6, T7, T8 각각에서, 보유하고 있는 모든 게이트에 고르게 출발 항공편이 배정된다.
- 항공사에 따른 차이는 고려하지 않는다.
- T1, T2, T3, T4, T5, T6, T7, T8 각각에서 보안 검색 위치는 한 곳씩 존재한다.

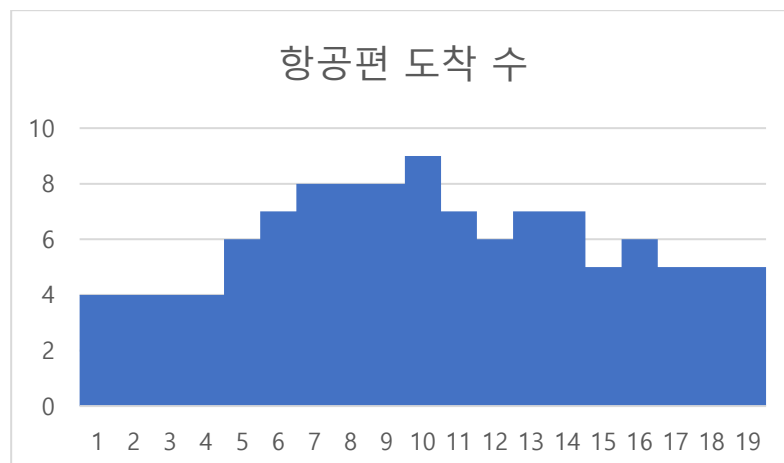
⁵ LAWA, <https://www.lawa.org/lawa-investor-relations/statistics-for-lax/volume-of-air-traffic>, 2023.12.15.

3. 데이터 가정

1) 승객 도착



- 하루 동안 국제선 항공편을 통해 TBIT로 도착하는 승객의 시간대별 인원 수는 위의 그래프와 같다.⁶ 2023년 10월 한 달, 31일의 데이터를 수집하여 평균을 낸 자료이며, X축은 시간대를 의미한다.(05시부터 24시까지 1시간 단위)
- 위와 같은 종 모양 분포를 확인하여, 승객의 도착이 중앙 시간대(12시~16시)에 증가하였다가 이후 다시 감소하는 것으로 추정하였다.
- 한 항공편에서 내리는 승객 간의 시간차는 경험적으로 보아 평균 4.3초로 가정한다.
- 같은 자료를 통해 확인한 시간대별 항공편 도착 수는 아래와 같다.



- 2023년 10월 기준, 하루에 평균적으로 도착하는 항공편은 115개이다.
- 따라서 각 게이트 당 평균 3.5개의 항공편이 도착하며, 모든 게이트

⁶ 미국 관세국경보호청, <https://awt.cbp.gov/>, 2023.12.15.

트는 최소 2대에서 최대 5대까지의 항공편을 수용한다.

2) 승객 별 환승 터미널 결정

- 아래의 표와 같이, 2023년 10월 각 터미널별 국내선 출발 승객 수 및 그 비율을 고려하였다.
- 해당 비율을 환승객들도 따를 것으로 가정하였다.

	국내선 출발 승객 수	비율(%)
T1	320,614	15
T2	282,988	14
T3	264,217	13
T4	191,880	9
T5	453,651	22
T6	140,203	7
T7	294,441	14
T8	131,834	6
합계	2,079,828	100

3) 이동시간 및 거리

- 모델에서 존재하는 모든 이동과 관련하여 도보 속력을 4km/h로 가정하고, 버스의 속력은 25km/h로 가정하였다.
- 이동거리는 구글맵을 활용하여 실측하였다. 실측 결과는 아래 표와 같다.

각 게이트 별 짐 찾는 곳까지의 거리							
Gate	130	131	132	133	134	135	137
거리(m)	130	130	180	180	220	270	270
Gate	139	141	148	150	151	152	153
거리(m)	300	300	70	170	170	240	240
Gate	154	155	156	157	159	201A	201B
거리(m)	320	320	410	410	410	370	370
Gate	202	203	204	205	206	208	209A
거리(m)	370	370	440	440	520	570	570
Gate	209B	210A	210B	221	225		
거리(m)	593	593	633	400	400		

입국장에서 각 터미널까지의 거리				
Terminal	T1	T2	T3	T4

거리(m) (버스)	1615	1875	2170	230
거리(m) (도보)	805	515	230	230
Terminal	T5	T6	T7	T8
거리(m)(버스)	1615	1875	2170	230
거리(m)(도보)	805	515	230	230

터미널 내부 길이				
Terminal	T1	T2	T3	T4
내부길이(m)	143	110	125	200
Terminal	T5	T6	T7	T8
내부길이(m)	240	233	200	195

4) 입국심사

- 소요시간은 1분 12초 이상 최대 3분 이하의 범위에서 scale이 2.3, shape이 2인 웨이블 분포를 따르는 것으로 가정하였다.
- 지문 인식 및 간단한 질문을 거치는 입국심사는 최소 1분 12초 이상 소요되고, 3분이 초과될 경우 이는 추가적인 면담을 행하는 경우로 가정하고 이는 곧 MCT를 충족하지 못하는 것으로 고려하여 제외하였다.
- 최대 66개의 입국심사 카운터를 운영할 수 있다고 가정하였다.

5) 짐 찾는 것에 소요되는 시간

- 분 기준, TRIA(3,5,10)의 삼각분포를 따르는 것으로 가정하였다.
- 10분 이상이 소요되는 경우에는 짐을 찾는 과정 중에 짐을 분실하는 등의 이상이 생긴 것으로 가정하고 이는 MCT를 충족하지 못하는 것으로 고려하여 제외하였다.

6) 환승 비율

- 모든 승객의 30%가 환승을 하는 것으로 가정하였다.

7) 신규 국내선 탑승객

- 한 달 기준 데이터인 3.2)의 표 국내선 출발 승객수를 각 터미널 별로 31(10월의 날짜수)로 나눈 수(이하 '총 출발승객 수')에서 3.B.를 통해 결정된 각 터미널 별 환승객 수를 제외(이하 '신규 출발승객 수')하였다.

- 3.1)에서 도출되는 각 시간대별 도착 승객 수 비율(도착 전체 승객 중 해당 시간대에서의 비율)을 통해 각 시간대별 터미널 별로 확인된 신규 국내선 탑승객 수를 추산하였다.
- 그 내용은 아래의 표와 같다.

신규 국내선 탑승객 수			
터미널	총 출발승객	환승객	신규 출발승객
T1	10,342	1145	9,197
T2	9,129	1069	8,060
T3	8,523	993	7,530
T4	6,190	687	5,503
T5	14,634	1680	12,954
T6	4,523	534	3,989
T7	9,498	1069	8,429
T8	4,253	458	3,795

시간대별 도착/출발 비율			
시간대	비율	시간대	비율
05-06	0.03	15-16	0.07
06-07	0.04	16-17	0.05
07-08	0.04	17-18	0.06
08-09	0.04	18-19	0.06
09-10	0.06	19-20	0.04
10-11	0.06	20-21	0.05
11-12	0.07	21-22	0.04
12-13	0.07	22-23	0.03
13-14	0.08	23-24	0.04
14-15	0.08		

시간대별 터미널별 신규 출발승객 수								
시간대	T1	T2	T3	T4	T5	T6	T7	T8
05-06	276	242	226	165	389	120	253	114
06-07	368	322	301	220	518	160	337	152
07-08	368	322	301	220	518	160	337	152
08-09	368	322	301	220	518	160	337	152
09-10	552	484	452	330	777	239	506	228
10-11	552	484	452	330	777	239	506	228
11-12	644	564	527	385	907	279	590	266

12-13	644	564	527	385	907	279	590	266
13-14	736	645	602	440	1036	319	674	304
14-15	736	645	602	440	1036	319	674	304
15-16	644	564	527	385	907	279	590	266
16-17	460	403	377	275	648	199	421	190
17-18	552	484	452	330	777	239	506	228
18-19	552	484	452	330	777	239	506	228
19-20	368	322	301	220	518	160	337	152
20-21	460	403	377	275	648	199	421	190
21-22	368	322	301	220	518	160	337	152
22-23	276	242	226	165	389	120	253	114
23-24	368	322	301	220	518	160	337	152

8) 체크인

- 체크인에 소요되는 시간은 분 기준 TRIA(1.3, 2, 2.3)을 따른다고 가정하였다.
- 체크인 카운터 관련, 3.3).의 표에서 각 터미널의 이용 비율 중 평균이 되는 T3(13%)에 60개의 카운터가 있는 것으로 가정하였다.
- 이를 기준으로 각 터미널에서의 체크인 카운터 수는 ' $30 \times (\text{해당터미널이용비율} / \text{T3이용비율})$ '로 설정하여, 이용객 수에 따라 카운터의 수를 설정하였다.

9) 보안검사

- 소요시간은 30초 이상 최대 2분 30초 이하의 범위에서 scale이 0.8, shape이 1.8인 웨이블 분포를 따르는 것으로 가정하였다.
- 각 터미널 별 보안검색대의 수는 위의 3.7)의 과정과 동일하게, T3의 검색대 수를 20개로 가정하고, 비율에 맞게 다른 터미널의 검색대 수를 설정하였다. 카운터 및 검색대 수는 아래의 표와 같다.

	T3에 대한 비율	카운터 수	검색대 수
T1	1.15	69	23
T2	1.08	65	22
T3(기준)	1	60	20
T4	0.69	41	14
T5	1.69	101	34
T6	0.54	32	11
T7	1.08	65	22
T8	0.46	28	9

4. 모델 세부 설명

1) PassengerArrival(Create)

- 이는 국제선 항공편을 통해 도착하는 승객(환승객 포함)(Passenger)을 생성하는 Create 모듈이다. 총 33개의 게이트에서 국제선 항공편이 도착하기에, 33개의 PassengerArrival Create 모듈이 존재한다.
- 이후에, 총 도착 인원을 통해 모델의 유효성을 판단할 것이다.

i. Schedule(Schedule)

- 이들 각각의 interarrival time에 Schedule을 활용하였다.
- 3.1)를 참고하여 Schedule을 설정하였다.
- 항공편이 도착하지 않는 시간의 Duration은 삼각분포를 활용하였고, 이에서 각 최빈값들을 기준으로 삼고 양쪽의 꼬리를 활용하여 3.1)의 분포를 반영하고자 하였다.
- 즉, 중앙시간대에서는 항공편이 도착하지 않는 시간을 짧게 함으로써, 해당 시간대에 보다 많은 항공편이 도착할 수 있도록 설계하였다.
- Value는 3.1)의 승객이 내리는 시간 차이에 대한 가정에 따라 설정하고 그 Duration을 약 13분에서 27분까지의 삼각분포로 설정하여, 각 항공편에서 내리는 승객의 수에 무작위성을 부여하였다.
- 해당 부분은 강의안 Ch.12 중 'Nonstationary Poisson Process'를 참고하였다.⁷

2) Assign(Assign)

- 각 Assign 모듈은 entity에 StartTime과 TerminalNum을 할당한다.
- StartTime(Attribute)에는 TNOW를 할당하여, 이후에 기록 과정에서 활용할 수 있도록 하였다.
- TerminalNum(Attribute)는 3.2)의 비율을 고려하여 DISC분포를 활용하였다.
- 이는 강의안 Ch.4의 Model 4-1을 참고하였다.

3) Gate(Station)

- 각 게이트를 표현하는 Station이다. 총 33개의 게이트를 표현하기 위해 33개의 Station 모듈이 존재한다.
- 모든 Station 모듈은 강의안 Ch.4의 Model 4-3을 참고하였다.

⁷ LAX 모델에서 사용된 모듈이 동일하게 DFW, SIN, ICN 모델에서도 사용되므로, 해당 모듈들을 사용하며 참고한 강의안 및 교재의 부분은 LAX 모델에서만 언급하겠다.

4) To CBP(Route)

- 짐 찾는 곳 및 입국 심사 장소까지 가는 이동 경로를 표현하는 Route이다.

i. Route Time

- 게이트의 위치에 따라 짐 찾는 곳 및 입국 심사 장소까지 소요되는 이동 시간을 다르게 책정하였다. 3.C.의 실측 거리 및 도보 속력을 활용하여 산출한 게이트 별 해당 이동에 소요되는 시간은 아래의 표와 같다.

- 따라서 각 게이트에서 CBP까지의 이동 시간은 아래의 소요시간을 최빈값으로 하고 이의 80%, 120% 각각을 최소값 및 최대값으로 하는 삼각분포를 따르도록 설정하였다.

- 모든 Route 모듈은 강의안 Ch.4의 Model 4-3을 참고하였다.

각 게이트 별 짐 찾는 곳까지의 소요시간							
Gate	130	131	132	133	134	135	137
분(m)	2	2	2.7	2.7	3.3	4.1	4.1
Gate	139	141	148	150	151	152	153
분(m)	4.5	4.5	1.1	2.6	2.6	3.6	3.6
Gate	154	155	156	157	159	201A	201B
분(m)	4.8	4.8	6.2	6.2	6.2	5.6	5.6
Gate	202	203	204	205	206	208	209A
분(m)	5.6	5.6	6.6	6.6	7.8	8.6	8.6
Gate	209B	210A	210B	221	225		
분(m)	8.9	8.9	9.5	6	6		

5) Customs and Border(Station)

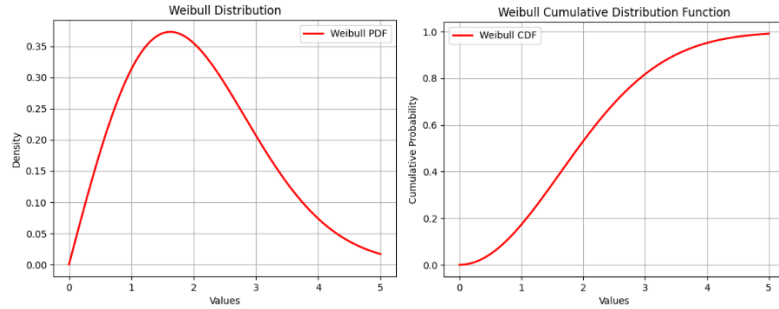
- 짐 찾는 곳과 입국 심사 장소가 있는 위치를 표현하는 Station이다. LAX에는 해당 장소가 한 곳으로 통합되어 있기에, 관련 Station도 하나 존재한다.

6) BorderProtection(Process)

- 입국 심사를 하는 과정을 표현하는 Process 모듈이다.

- 3.4)의 가정에 따라 소요시간을 표현하였다.

- 사용된 분포의 PDF와 CDF는 아래의 그림과 같다.



i. PassportControl(Resource)

- 입국 심사 장소 자체는 한 곳이기때문에, 오직 하나의 관련된 Resource가 존재한다.

1. PassportOfficer(Schedule)

- 해당 Resource의 Capacity를 Schedule을 통해 설정하였다. 이는 3.4)에서의 최대 운용가능한 카운터 수(66개) 및 도착 승객이 중앙 시간대에서 증가하는 것을 고려하여, 중간 시간대까지 Capacity 또한 점차 증가한 뒤 다시 감소하는 형태로 설정하였다.
- 이는 Ch.4의 Model 4-2를 참고하였다.

7) Baggage Claim(Process)

- 짐 찾는 것에 소요되는 시간을 표현하기 위한 Process이다. 그 과정이 특별히 특정 Resource를 seize하고 대기열이 발생하는 것은 아니기에, Delay로만 표현하였다.
- 소요 시간은 3.5)의 가정에 따라 설정하였다.

8) RecordTime at CBP(Record)

- 입국 과정을 모두 마칠 때까지 소요된 시간을 중간 기록하기 위한 Record 모듈이다.
- Time Interval을 통해 이전 과정에서 할당된 StartTime의 TNOW값과의 차이를 기록하였다.
- 이를 포함한 모든 Record 모듈은 Ch.4의 Model 4-1을 참고하였다.

9) Decide transfer(Decide)

- 3.6)의 가정에 따라, 2-way by chance를 통해 True를 70%로 설정하여 전체 도착 승객 중 30%만이 환승할 게이트로 이동하도록 하였다.
- 모든 Decide 모듈은 Ch.4의 Model 4-1을 참고하였다.

10) Decide terminal(Decide)

- 환승객 각자에게 배정된 환승할 터미널에 따라 경로를 분리하는 Decide 모듈이다.
- N-way by condition을 활용하여 기존 attribute인 TerminalNum의 값에 따라 경로가 결정되도록 설정하였다.

11) To Terminal(Route)

- 입국한 TBIT에서 환승할 터미널(T1, T2, T3, T4, T5, T6, T7, T8)로 이동하는 경로를 나타낸다.
- 추가로, 이에서 강의안 Ch.8의 Transporter를 고려해보았으나, 이 또한 버스에 탈 수 있는 최대 인원을 반영하기 어렵다는 점에서 계획대로 Station-Route를 활용하였다.

i. Route Time

- 일부 환승객은 환승할 터미널 및 버스 대기시간에 따라, 버스를 이용할지 혹은 도보로 이동할지 선택할 수 있다.
- 모델에는 해당 두가지 방법 중 최소시간이 소요되는 방법을 택하는 것으로 설정하였다.
- 버스를 이용한 이동시간은 3.3)에서 가정한 거리와 버스의 속력을 통해 산출하였으며, 모델에는 이의 90%와 110%를 추가하여 삼각분포로 반영하였다. 추가로, 버스를 대기하는 시간을 반영하기 위해 버스 이동시간에 대기시간 UNIF(0,3)을 더하였다.
- 도보를 통한 이동시간 또한 동일하게 3.3)의 거리 및 속력을 통해 산출하였고, 이의 90%와 110%를 추가하여 삼각분포로 반영하였다.
- 이와 같은 상황에서, 대부분의 경우는 버스(T1, T5, T6, T7, T8) 또는 도보(T3) 중 한가지가 항상 빠른 것으로 나타나나, T2과 T4로 가는 경우에는 양자 중 항상 빠른 경우가 특정되지 않는다.
- 따라서 T2와 T4에 대해서는 MN()을 통해 두가지 경로 중 한가지를 특성에 따라 선택할 수 있도록 설정하였고, 나머지 경로에 대해서는 해당되는 최소 시간을 설정하였다.
- 3.3)를 통해 산출한 이동시간은 아래와 같다. 모델에서의 이동시간은 Expression을 통해 표현하였다.

Terminal	T1	T2	T3	T4
분(m)(버스)	3.9	4.5	5.2	0.6
90%	3.5	4.1	4.7	0.5
110%	4.3	5	5.7	0.7
분(m)(도보)	T1	T2	T3	T4
분(m)(도보)	12.1	7.7	3.5	3.5
90%	10.9	6.9	3.2	3.2
110%	13.3	8.5	3.9	3.9
Terminal	T5	T6	T7	T8
분(m)(버스)	1	1.5	2.2	4.8
90%	0.9	1.4	2	4.3

110%	1.1	1.7	2.4	5.3
분(m)(도보)	6.3	9.5	13.7	16.3
90%	5.7	8.6	12.3	14.7
110%	6.9	10.5	15.1	17.9

12) Terminal(Station)

- 각 터미널을 나타내는 Station이다.

13) NewPassenger(Create)

- 각 터미널에 국내선 항공편 탑승을 위해 새롭게 도착하는 승객 (NewPassenger)을 생성하기 위한 Create이다.

i. NewArrival(Schedule)

- 시간에 따른 혼잡도를 고려하기 위해 3.7)의 시간대별 터미널 별 신규 출발승객 수 표를 그 자체로 스케줄을 부여함으로써 실제와 가까운 arrival rate을 표현하였다.
- 해당 부분은 강의안 Ch.12 중 'Nonstationary Poisson Process'를 참고하였다.

14) Ticketing and Bag drop(Process)

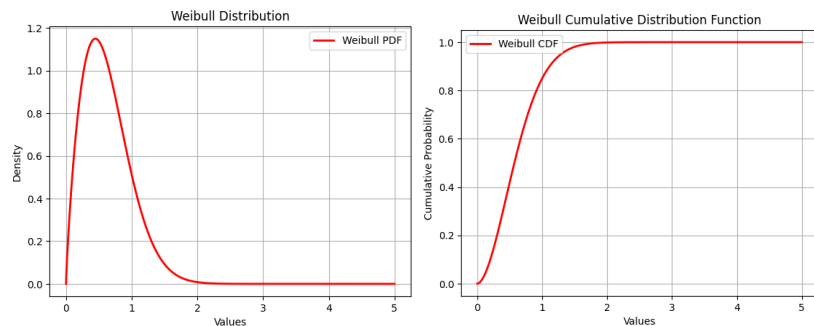
- 체크인에 해당하는 Process로, 3.8)에 따라 소요시간을 표현하였다.

i. CheckIn(Resource)

- 항공사는 모델에서 고려하지 않음에 따라, 터미널 당 Resource는 오직 하나가 존재한다.
- 해당 Resource의 capacity는 터미널 별로 상이하며, 이는 3.9)의 표와 같다.

15) Security(Process)

- 보안 검색에 해당하는 Process로, 3.9)에 따라 소요시간을 표현하였다. 분포를 나타내면 아래의 그림과 같다.



i. Security(Resource)

- 터미널 당 보안 검색 위치가 한 곳씩 존재한다는 구조 가정에 따라, 각 터미널에는 하나의 보안 검색 Resource가 존재한다.
- 해당 Resource의 capacity는 터미널 별로 상이하며, 이는 3.9)

의 표와 같다.

16) WalkingTime(Process)

- 터미널 내에서 환승할 게이트까지 이동하는 시간을 표현하는 Process이며, 별도의 Resource 및 대기열이 없기에 Delay로 표현하였다.
- 3.3)에서의 터미널 내부 길이에 따른 터미널 내 최대 이동시간은 아래의 표와 같다.

터미널 내 최대 이동시간				
Terminal	T1	T2	T3	T4
내부길이(m)	2.1	1.7	1.9	3
Terminal	T5	T6	T7	T8
내부길이(m)	3.6	3.5	3	2.9

- 모델에서는 UNIF(0,(각 터미널에서의 최대 이동시간))으로 반영하여, 터미널 내에서의 이동에 소요되는 시간을 설정하였다.

17) RecordTime(Record)

- 환승 과정을 모두 마칠 때까지 최종적으로 소요된 시간을 기록하기 위한 Record 모듈이다.
- Time Interval을 통해 이전에 할당된 StartTime의 TNOW값과의 차이를 기록하였다.

18) WriteOutput(Readwrite)

- 결과 분석을 위해 output을 csv(LAX_output.csv)로 내보내는 Readwrite 모듈이다.
- 이에서 컬럼 값을 replication number(NREP)와 총 환승 소요 시간(TNOW-StartTime)으로 설정하고, File(data module) 모듈에서 파일 이름 등을 지정하였다.

19) Leave, TransferGate(Dispose)

- Leave는 입국 직후, 환승을 하지 않는 승객을 방출하는 Dispose 모듈이다. 이로 방출되는 승객은 결과에서 고려되지 않아야 하기에, 모듈 내에서 'Record Entity Statistics'를 해제하였다.
- TransferGate는 환승객이 환승하는 최종 환승 게이트를 의미하는 Dispose 모듈이다.

5. 모델 검증 계획

- 총 30번의 replication 동안 도착하는 승객(Passenger)의 NumberIN의 평균이 실제 하루동안 도착하는 국제선 항공편 승객 수인 25450명(3.1)을 통해 도출)과 유사하면 모델이 검증된 것으로 판단할 것이다.

- 구체적으로, NumberIN이 [실제승객수*0.95, 실제승객수*1.05]의 구간 내에 있는 경우 모델이 실제와 유사하게 설계되었다고 판단할 것이다.

C. 댈러스-포트워스 국제공항(Dallas-FortWorth International Airport, DFW)

i. 공항설명

2022년 기준, DFW 국제공항은 탑승객 수를 기준으로 세계에서 두 번째로 바쁜 공항으로 기록되었다.⁸ 또한 규모에서도 미국에서 두 번째 크기를 기록할 정도로 가장 큰 규모를 갖고 있다.

1. 공항구조

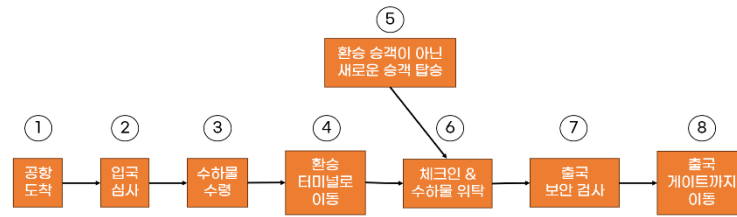
- DFW 국제공항은 총 5개의 터미널(터미널 A, B, C, D, E)로 이루어져 있으며 한 개의 국제터미널(터미널 D, E), 세 개의 국내터미널로 이루어져 있다. 각 터미널은 모두 반원 모양이지만 크기에 차이가 있으며, 입/출국 게이트 수나 출국을 위한 보안검사대의 숫자 등에 차이가 존재한다.
- 터미널 간의 이동은 걷거나 공항 내 교통 수단을 이용하는 것을 통해 가능하며, 교통 수단으로는 Secured Area 내에서는 Skylink(기차), 밖에서는 Terminal Link(버스)를 이용할 수 있다.
- 승객의 도착을 시뮬레이션하는 터미널 D는 총 32개의 게이트로 이루어져 있으며, 별도의 출국 게이트에 대한 자료가 없어 동일한 게이트에서 출입국이 별도로 이루어지는 것으로 간주하였다.
- 나머지 터미널 A, B, C, E는 각각 26, 33, 29, 29개의 게이트로 이루어져 있다.
- 보안 검사를 거치는 TSA Security Gate는 터미널 A-E 순서로 각각 3, 2, 3, 3, 4개를 갖고 있다.

2. 특이사항

- 터미널 E는 미국과 캐나다 편 항공기만이 이용하므로 본 프로젝트에서는 국내선 터미널로 간주하였다.
- 본 시뮬레이션은 국제선으로 도착한 후의 환승을 기준으로 하므로, 환승객들은 환승 전에 항상 수하물을 수령한 뒤 새롭게 체크인을 해야 하는 경우만을 고려하였다. 또한 수하물 수령은 터미널 D의 Secured Area 외부에 존재하는 특수 구역에서만 가능하다.

⁸ [ACI World confirms top 20 busiest airports worldwide | ACI World](#)

ii. 시뮬레이션 모델설명



아래나 모델로 구현한 시뮬레이션 모델의 전반적인 과정에 대한 설명이다. 아래 번호는 위 그림에서 부여된 기준으로 하여 지정하였다.

1. 구조 및 기본 가정`

1) 공항 도착

항공기가 공항에 도착한 후에 승객이 내려 게이트까지 가는 과정은 따로 고려하지 않았다. 공항에 도착한 직후에는 곧바로 게이트에 도달하는 것으로 가정했다. 관련된 정확한 데이터가 부재하여 이를 고려하기 어려웠기에, 시뮬레이션 과정에 직접적으로 구현하지는 않았다. 이 부분에 대해서는 실험 결과를 바탕으로 MCT를 검증하는 과정에서 해당 시간에 대한 경험적인 수치를 고려하고자 하였다.

2) 입국 심사

게이트 별로 존재하는 입국심사대를 통과한다. 또한, 입국 심사 과정에서 과도하게 시간이 소요되어서 환승을 하지 못하는 경우는 MCT 설정에 있어서 유의미하게 고려할 필요가 없다고 판단하여 고려하지 않았다.

3) 수하물 수령

수하물을 수령하기 위해서는 도착 터미널인 터미널 D에서 수하물 수령위치까지 걸어가야 한다.

4) 수하물 수령 후 환승 터미널로 이동하는 경우는

- **터미널 D에서 보안 검사를 실시하고** Skylink를 이용해 환승 터미널로 이동하는 경우
 - **터미널 D에서 보안검사를 실시하지 않고** Terminal Link를 이용해서 환승 터미널로 이동하는 경우
- 의 두 가지로 나뉘어진다.

본 시뮬레이션에서는 가정의 단순화의 목적을 위해서, 그리고 두 가지 경우의 선호도에 대한 데이터가 존재하지 않는 문제로 두 번째 경우(터미널 D에서 보안검사를 실시하지 않는 경우)에 대해서만 고려하였다. 위의 설명대로 터미널 간 Terminal Link를 이용하는 경우에는 상정하였으며, 10분 간격의 배차가 있기 때문에 이를 고려하였다. 이동하는 과정에서는 대기열이 발생하지 않고, 각자가 독립적으로 걸어서

Terminal Link의 정류장까지 이동하는 상황을 상정했다. 터미널을 이동한 후에 새롭게 탑승하는 승객들과 섞여 체크인을 하고 보안검사를 하는 부분을 고려하였다.

5) 환승 승객이 아닌 새로운 승객 탑승

시뮬레이션 과정에서 공항의 실제 혼잡한 상황을 정확히 모사하기 위해 DFW 공항에서 출발하는 항공기를 탑승하고자 하는 승객들을 고려하였다.

6) 체크인 & 수하물 위탁

별도의 큐를 만들지 않고, 승객들이 모두 독립적으로 체크인과 수하물 위탁에 시간이 소요되는 것으로 가정하였다.

7) 출국 보안 검사

공항 구조 *설명*에서 언급하였던 터미널 별로 존재하는 TSA Security Gate를 이용한다.

8) 출국 게이트까지 이동

별다른 특이사항 없이 출국을 하고자 하는 비행기에 탑승하는 게이트까지 걸어서 이동한다. 이렇게 출국 게이트까지 이동하면 환승을 마친 것으로 보았다.

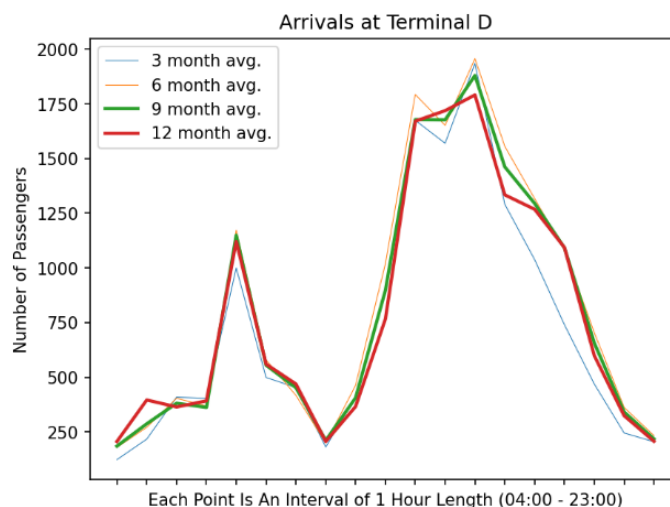
2. 구조 가정

- 공항의 운영 시간은 오전 4시에서 오후 10시까지로 한다. 이는 공항의 각 TSA Security의 운영 시간을 고려하고, CBP의 승객 데이터가 존재하는 시점을 기준으로 하였다.

3. 데이터 가정

앞서 설명한 시뮬레이션 모델의 각 과정에서 이용된 수치에 대한 설명이다.

1) 터미널 D에 도착하는 승객



공항에 도착한 승객들의 수는 [U.S Customs and Border Protection](#)(CBP)

의 도착 승객 데이터를 기준으로 운영 시간 동안의 1시간 단위 최근 3, 6, 9, 12개월 평균을 구하였다. (위 그림) 이들 평균 값 중 9개월 평균이 가장 시간대별 변동폭을 적절히 반영하면서 과도한 이상치가 존재하지 않다고 생각하여 9개월 평균 값이 나타내는 시간대별 비율을 이용해서 시간 대별 탑승 승객의 숫자를 가정하였다. 도착하는 승객은 항공기 도착 후 한 번에 생겨난다는 점을 고려하여 항공기의 대수를 기준으로 승객의 수를 추정하고자 했다.

터미널 별로 도착하는 항공기의 대수를 추정하는 과정은 아래 3.5에 상세하게 서술하였다. 다만, 터미널 D에 도착하는 승객의 시간대별 수치는 CBP에서 제공하는 정확한 수치를 기반으로 15분 단위가 아니라 1시간 단위로 추정하였다. (즉, 3.5의 5-7 과정 대신, CBP에서 제공하는 데이터를 직접적으로 이용하여 - 3.5의 4번 과정과 함께 - 얻은 아래 그림의 데이터를 비율로 계산하여 각 시간대별 항공편의 도착 수의 비율을 추정하였다) 추가적으로, 상황의 단순화를 위해서 환승 승객이 발생하는 터미널 D에 도착하는 승객들은 존재하는 무작위로 32개의 게이트를 이용한다고 가정했다. (이 과정으로 구한 수치는 csv 파일(d_arrival.csv)로 첨부하였다.)

- 2) **입국 심사 과정**은 웨이블 분포를 이용하였고, 이는 입국심사에 걸리는 소요시간의 분포가 최솟값에 대해서 Skewed 되어있을 것이라 추측하였기 때문이다. 또한, scale과 shape의 파라미터를 조정하여 적절한 분포를 찾았으며, 최소/평균/최대 소요시간의 통계⁹를 참조하여 실제 Wait Time과 유사한 평균값이 나오도록 Resource의 Capacity를 조정하며 실제 상황과 유사하게 시뮬레이션 하고자 하였다. 또한, 분포의 최솟값에 제한을 두어서 과도하게 짧게 시간이 소요되는 것을 막았다. 마찬가지로 최댓값에도 제한을 두어 실제 소요시간과 괴리되어 대기 시간이 비정상적으로 길어지는 것을 방지하였다.
- 3) **수하물을 수령하기 위해 이동하고 대기하는 시간**은 적절한 데이터를 찾기 어려워 평균 걷는 시간과 경험적인 대기시간을 바탕으로 한 삼각분포를 이용하였다.
- 4) **터미널 간 이동 시간**은 데이터를 기준으로 상수 값으로 구현했고, 터미널 D에서 D로 이동하는 경우에는 평균 걷는 속도를 바탕으로 삼각분포를 이용하였다.
- 5) 최대한 정확한 **탑승객의 숫자를 추정**하기 위해서 아래와 같은 과정을 거쳤다. 최종적으로 구한 수치에서 소수점 둘째 자리까지 반올림

⁹ [Average Immigration & Customs Wait Times by U.S. Airport \[2023\] \(upgradedpoints.com\)](https://www.upgradedpoints.com/average-immigration-customs-wait-times-by-u.s.-airport-2023/)

하여 실제 시뮬레이션 실험에 이용하였다.

- ① [Statista](#)¹⁰에서 DFW 국제공항의 Operation 데이터를 바탕으로 연간 배정되는 항공편의 수에 대한 지난 10년 간의 평균치를 구하였다. (이상치를 제외하기 위해 코로나에 영향을 크게 받은 2020, 2021년은 제외함): 연간 약 670편의 항공편
- ② 위의 과정에서 얻은 연간 데이터를 12로 나눈 뒤 다시 30으로 나누어 하루 동안의 출발/도착 항공편의 수를 구하였다:
일일 약 1860대의 항공편
- ③ 이렇게 얻은 하루 간의 항공편 수를 DFW 국제공항의 공식 데이터¹¹로 얻을 수 있는 2023년 1월-10월 간의 출발/도착하는 승객의 비율을 바탕으로 “출발” 하는 항공편의 대수를 추정하였다.
(약 5:5)
- ④ **터미널 D에 대한 출발/도착 항공편의 수**는 아래와 같은 과정으로 추정하였다.
먼저 실제 CBP 데이터와 터미널 D의 도착 승객 수의 데이터를 일치시키기 위해서 DFW 국제 공항의 공식 홈페이지¹²에서 제공하는 국제/국내 승객의 비율을 이용하였다. 본 데이터는 승객의 국적의 데이터이기에 항공기의 국적과는 무관하지만, 이를 이용한 비율을 산출했을 때 CBP로 얻은 일일 평균 수치와 대동소이한 수치를 보여 시뮬레이션 과정에 이용 가능함을 검증하였다. 사용된 수치의 일관성을 위해 CBP 데이터를 새롭게 이용하지 않았다.
CBP 데이터로 얻은 터미널 D의 일일평균 도착 승객 수(**15186**), 전체 항공기 대수 1860과 International 승객 수의 비율 0.1363을 곱하고 2로 나눈 뒤 항공기 평균 도착 수를 곱하여 도착 승객 수를 표현한 결과(약 **15465**)
- ⑤ 3의 과정으로 얻은 출발 항공편의 수를 이용해, DFW 국제공항의 공식 홈페이지¹³에서 얻을 수 있는 실시간 TSA Wait Time 값을 바탕으로 **각 국내 터미널(A, B, C, E) 별로 출발하는 항공편의 수**

¹⁰ [Flight operations at Dallas/Fort Worth International Airport 2022 | Statista](#)

¹¹ [DFW International Airport | Official Website \(dfwairport.com\)](#)

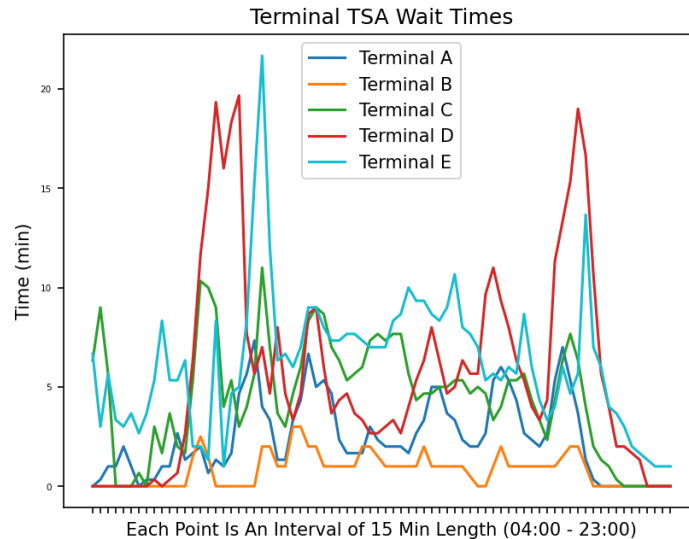
¹² [DFW International Airport | Official Website \(dfwairport.com\)](#)

¹³ [DFW offers multiple checkpoints at key locations in each terminal. View wait times, read security tips and find more info on TSA PreCheck. \(dfwairport.com\)](#)

를 추정하였다. TSA Wait Time이 길다는 것은 곧 터미널이 혼잡하다는 의미이며, 터미널이 혼잡하다는 것은 곧 많은 사람이 이용한다는 것이라는 점에서 착안하였다. 따라서, 출발 항공편에 탑승을 원하는 승객의 수를 추정하기 위해서 TSA Wait Time을 이용하였다.

⑥ TSA Wait Time은 다음과 같은 과정으로 구하였다.

- 각 터미널 별로 존재하는 여러 개의 Security Check에 대한 Wait Time 각각에 대한 수치를 DFW 국제공항의 공식 홈페이지를 통해 얻었다.
- 이를 각 TSA Security Check Wait Time을 터미널 별로 평균 낸 뒤, 각 터미널의 전체 Wait Time의 평균을 구하였다. (아래 그림)



평균값: [A, B, C, D, E] = [2.41, 0.8, 4.49, 5.73, 6.25] (단위: 분)

- 이를 바탕으로 총 5개의 터미널의 Wait Time에 대한 상대적인 비율을 구하였다. 이후 3번 과정에서 얻은 DFW 국제공항에서 출발하는 항공편의 수에 각각 곱해, 각 터미널에서 출발하는 항공편의 대수를 추정하였다.

비율: [A, B, C, D, E] = [0.12, 0.04, 0.23, 0.29, 0.32]

⑦ 이후 추정한 터미널 별 항공편 수의 비율을 바탕으로 시간대별 출발 항공편 이용 승객의 숫자를 추정하였다. 시간대별 이용 승객은 4번의 과정과 마찬가지로, 터미널 별 Security Wait Time의 평균을 이용하여 시간 대별로 도착하는 항공편의 대수를 추정하였다.

⑧ Security Wait Time이 15분 단위로 등록되어 있었기 때문에, 위 과정들로부터 각 터미널에 대해서 출발하는 항공편의 대수를 15

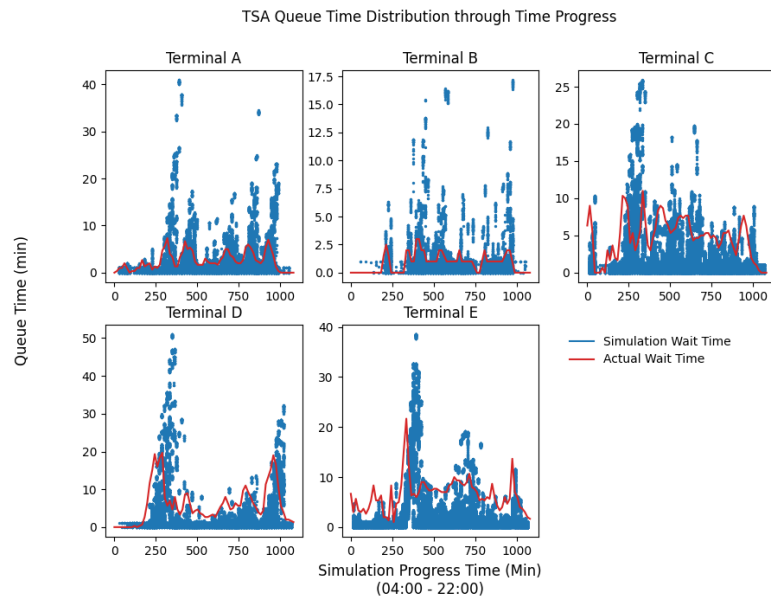
분 단위로 추정하였다. 이렇게 매 15분 간격으로 출발 항공편의 숫자를 최종 추정하였다.

- ⑨ 또한 환승하는 승객들도 탑승하는 승객 수 데이터에 포함되어 있는 점을 고려하였다. 국제 터미널인 터미널 D에 내리는 승객들 중 40%가 환승을 한다는 점을 가정하고, 이들 환승 승객의 수를 터미널 Security Wait Time을 기준으로 나누어 7번까지의 과정으로 구한 터미널 별 출발 승객 숫자에서 감산(Subtract)하였다. (이렇게 구한 구체적인 수치는 csv 파일(net_a, net_b, net_c, net_d, net_e)로 첨부하였다, net_뒤의 알파벳은 터미널의 명칭을 나타냄)
- ⑩ 정리하자면 DFW 국제공항에서 하루 동안 출발하는 항공편의 대수가 약 1860개, 그리고 하루 동안 DFW 공항을 이용하는 승객들의 숫자는 약 226,520명으로 추정되었다. 따라서 이를 바탕으로 항공기 한 대당 약 122명의 승객이 탑승하고 있는 것으로 가정하였다. 최종적으로 Create 모듈에서 항공편의 도착 수를 Arrival Rate로 조정하고, Entity Number Per Arrival을 항공기 당 탑승 승객의 수인 122으로 조정하여 한 번에 승객들이 밀려오는 것을 구현하였다.

- 6) 체크인, 수하물을 위탁하는 과정 또한 구체적인 데이터를 찾기 어려워 경험적인 수치를 바탕으로 한 삼각분포를 가정하였다. 시간이 많이 소요되는 환승 여부에 영향을 미치는 것은 보안검사와 입국심사의 대기 시간임을 고려해서, 이 부분에서는 별도로 큐를 만들지 않고 Delay 모듈을 이용해 처리하였다.
- 7) 대부분의 경우 출국 보안 검사의 과정 자체에는 시간이 오래 걸리지 않는 점과 과정의 대부분이 짧은 시간이라는 점을 바탕으로 웨이블 분포를 이용하였다. 또한 입국 심사 과정에서와 마찬가지로 최소값에 제한을 두어서 과도하게 짧게 시간이 소요되지 않아야 하는 점을 고려하였다. 마찬가지로 최대값에도 제한을 두어 실제 소요시간과 괴리되어 소요시간이 비정상적으로 길어지지 않도록 해야 한다는 점 또한 고려하였다.

또한 시뮬레이션을 반복적으로 진행하면서, 각 큐에서의 대기 시간이 공항의 공식 홈페이지에서 추출한 각 터미널 별 최근 24시간 사이클과 유사한 양상이 되도록 조정하였다. 즉, 앞서 정의한 데이터들과 승객의 출발/도착에 관한 추정된 수치들이 Security Wait Time 데이터를 기반으로 한 공항의 혼잡도를 시뮬레이션 과정에서 반영하고 있는지 검증하였다. 이를 시뮬레이션 시간에 따라 시계열 분포로 시각

화 하여 검증한 결과, 실제 데이터를 기반으로 한 Security Wait Time 과 시뮬레이션을 통해서 얻은 Queue Time이 시간대별로 유사한 사이클 양상을 띄며 실제 공항의 시간대별 혼잡대를 반영한다고 판단하였다. (아래 그림)



- 8) 최종 게이트까지 이동하는 과정에는, 최종 탑승하고자 하는 게이트를 별도로 지정하지 않고, TSA Security Check에서 게이트까지의 거리를 바탕으로 평균 걷기 속도를 반영하여 Delay 모듈을 이용했다. 아무리 혼잡하더라도 평균적인 속도로 걸을 수 있다고 판단하였으며, 실제로 공항의 혼잡한 상황이 게이트까지 이동하는 것에 방해를 일으키는 경우가 거의 없다는 점을 고려하였다.

4. 모델 세부 설명

위의 과정으로 추정된 수치들을 바탕으로 구현한 시뮬레이션의 각 과정에서 이용된 모듈에 대한 세부적인 설명이다. 순서는 최초에 제시한 시뮬레이션 모델에 나타나있는 번호의 순서를 따랐다.

- 1) 사용한 모듈: Create(Entity: Transfer)

→ Decide

→ Assign(Attribute, Arrival Time = TNOW)

전체 터미널 D에 도착하는 항공기들을 Create 모듈로 Arrival 당 batch size를 122로 지정해 생성한 뒤, Decide 모듈을 이용해 무작위적으로 32개의 게이트에 승객들이 이동하도록 하였다. 항공기가 도착하는 지점 근처의 게이트들로 한 번에 승객이 이동하는 것이 좀 더 정확하다 볼 수 있을 것이다. 그러나 본 시뮬레이션에서는 최초 항공기 도착 이후 게이트까지의 이동은 유의미한 요소가 아니라 하였기

에 제외하였다. 이에 더해 항공기 도착 후 승객이 향하는 게이트를 임의로 배정함으로써 공항 내부 상황의 변동성을 표현하고자 하였다.

또한 직후 Assign 모듈을 활용하여 Arrival Time을 Attribute로 TNOW 수치를 부여함으로 최종적으로 환승 시간을 기록하는 데에 이용하였다.

- 2) 사용한 모듈: Process(Seize – Delay – Release, MN(MX(WEIB(2.3,2),2),5))
→ Decide (→ Dispose(환승객이 아닌 경우) OR → Decide(환승 터미널 배정))

입국 심사 과정은 큐를 갖는 Process 모듈을 이용했다. 각 게이트 별로 3개의 Capacity를 갖는 Resource를 정의하여 게이트 별 입국심사 카운터를 구현하였다. Resource의 Capacity는 시뮬레이션을 반복적으로 시도해가면서 Custom의 Queue에서의 Wait Time의 평균값이 과도하게 만들어지지 않도록 조정하여 얻은 값이다

입국 심사 이후에 모두가 환승을 하는 것은 아니므로 전체 도착 승객의 일부가 환승하도록 Decide 모듈을 이용했다. 환승하지 않는 승객은 Dispose 모듈을 이용해 시뮬레이션 과정에서 빠져나가도록 했다. Decide 모듈에서 환승 승객의 비율은 40%으로 정하였으며, 이는 **정확한 데이터를 찾기가 어려워 임의로 정하였다.**

위의 과정 이후에는 3.5.4의 과정에서 구한 Wait Time의 터미널 별 비율을 이용한 Decide 모듈을 이용해 환승 승객이 환승하는 터미널을 임의적으로 배정한 뒤 Assign 모듈을 이용해 환승 터미널을 Attribute로 부여하였다.

- 3) 사용한 모듈: Delay(TRIA(1, 5, 10)) → Delay(TRIA(2, 3, 20))

수하물 수령을 하기 위해 **이동하는 과정**과 수하물 수령을 위한 대기하는 시간 모두 따로 Queue가 존재하지 않는 Delay 모듈을 이용하였다. 그 이유는 본 시뮬레이션에서 예측할 수 없는 변동성이 유의미하게 작용하기 않다고 생각하였고, 환승 과정에서 예기치 못하게 시간을 소비하는 가장 중요한 과정은 입국/출국 심사 과정이라 판단하였기 때문이다. 수하물을 수령하는 구역까지 이동하는 것은 거리를 바탕으로 수치를 부여하였으며, 수하물 수령을 하기 위해 대기하는 시간은 경험적인 수치를 바탕으로 최대/최소값을 지정하여 두 가지 모두 삼각분포를 이용하였다.

- 4) Decide(Attribute 바탕 환승 터미널 별 이동 배정) → Route

앞서 2)번 과정에서 부여된 환승 터미널의 정보가 담긴 Attribute를 바탕으로 Decide 모듈을 이용해 환승 터미널로 이동하도록 하였다. 이 때, Route와 Station 모듈을 이용해 <전체 입국 후 수속 과정>과 <환승을 위한 과정>을 시각적으로 분리하여 시뮬레이션을 진행하고자 하였다. 또한, 10분 간격으로 배차가 되는 Terminal Link를 Route의 Route Time을 현재시간인 TNOW를 10으로 나눈 나머지를 기준으로 판단하는 방법으로 구현하였다.

Route 모듈의 Route Time을 이용해서 환승터미널까지 이동하는 시간을 고려하였다. 세부적인 Expression은 아래와 같다.

- ① 터미널 A: $TRIA(4, 6, 10)$, 걸기 때문에 삼각분포로 개인마다 편차가 있는 점을 고려함
- ② 터미널 B – D: modular 함수로 10분간의 배차간격을 고려하고 터미널 간 Terminal Link(버스)의 이동시간과 버스의 탑승장까지 이동시간을 고려하였다.
 - ✓ 터미널 B: $(10 - MOD(TNOW, 10)) + 5 + TRIA(2, 3, 5)$
 - ✓ 터미널 C: $(10 - MOD(TNOW, 10)) + 8 + TRIA(2, 3, 5)$
 - ✓ 터미널 D: $(10 - MOD(TNOW, 10)) + 8 + TRIA(2, 3, 5)$
 - ✓ 터미널 E: $(10 - MOD(TNOW, 10)) + 10 + TRIA(2, 3, 5)$

- 5) 사용한 모듈: Create → Record(터미널 별 환승하는 승객의 숫자)
또한 터미널에 도착한 후에 환승객 외에도 DFW 공항에서 처음 탑승하는 승객들을 새로운 Entity(New Passenger)로 Create하여 공항의 실제 상황을 구현하고자 했다. Create의 Arrival Rate의 수치는 3.5의 과정으로 얻은 출발 항공편의 수치를 1시간 단위로 조정하여 이용하였다. 또한 이 과정에서 새롭게 탑승하는 승객의 숫자는, 터미널 D에서 내려서 환승하는 승객들의 숫자를 감안하여 조정하였다.

위 과정에서 Record 모듈을 이용해 터미널에서 환승을 시도하는 사람들의 숫자를 기록하였고, 마지막에 Dispose이전 Record 모듈을 환승 시간에 대한 Tally를 기록하였다. 또한, 시뮬레이션이 올바르게 진행되고 있는지 타당성 검증을 하였으며, 각 entity의 Number In, Number Out의 수치가 실제 이용한 데이터와 유사한 값으로 나타났다:

실제 데이터를 기반으로 한 DFW 공항의 도착 승객, 공항 출발 승객의 추정값은 각각 112919.54, 15186이며 시뮬레이션으로 얻은 entity의 발생 수는 각 경우에 대해 109434.50, 15230이었다. 도착 승객의 데이터의 경우에 터미널 D에서 환승하는 승객을 위해서 감산하여 arrival rate을 정한 점을 생각하면 3%의 차이는 무시할 수 있다고 판단할 수 있다.

- 6) Delay: TRIA(3, 3, 10)
 3)번과 유사한 이유로 체크인 & 수하물 위탁 과정은 별도의 Queue가 없는 Delay 모듈로 구현하였다. 참고적으로, Queue를 갖는 Process 모듈을 사용하기 위해 Check In에 대한 Resource Capacity를 실제 공항에 있는 Check In의 수를 항공사를 고려하지 않고 일괄적으로 합산한 것을 바탕으로 이용해보았다. 그러나 이 과정에서 Resource의 Capacity가 충분치 못하다는 점을 발견했고, 시뮬레이션 과정에서 Check In에 대한 큐까지는 고려하기가 어렵다는 점을 발견하여 해당 과정은 Queue가 없는 Delay 모듈을 이용하여 구현하였다. 또한, 과도하게 기다리는 시간을 만들지 않는 선에서 일반적으로 작업을 수행하는데 걸리는 시간을 고려하였다. 추가적으로 최근에는 모바일 등으로 체크인을 하는 점 등을 고려하여 경험적으로 수치를 선정하여 삼각분포로 표현하였다.
- 7) 사용한 모듈: Process(Seize – Delay – Release, MN(MX(WEIB(2.3,2),1),2))
 → Assign(Attribute, TSA Queue Time = TNOW)
 → Decide(AINT(MOD(TNOW, 15)) == 0)
 → Assign(Attribute, TSA Queue Time = TNOW – TSA Queue Time)
 → Assign(Variable, Current Queue Time)
 → ReadWrite(Current Queue Time)
 출국 보안 검사 과정은 입국 심사와 동일하게 큐를 구현한 Process 모듈을 이용했다. 다만, Resource의 Capacity는 각 TSA Security Check의 Opening/Closing Time이 각 Security Check를 이용하는 승객의 숫자를 반영하며, 따라서 각각의 크기를 반영한다고 가정하여 이를 감안하여 숫자를 설정하였다.
 또한 시뮬레이션을 반복적으로 진행하면서, 각 큐에서의 대기 시간이 공항의 공식 홈페이지에서 추출한 최근 24시간 대기시간에 대한 평균값과 유사한 값이 되도록 Capacity를 조정하였다. 입국 심사별 Resource에 대해서는 Set을 이용하였으며, 공항의 크기가 커 입국 심사의 대기열과는 상관 없이 가까운 지점의 TSA를 이용할 것이라는

가정과 함께, 이를 임의로 지정하기 어렵기 때문에 Set의 Resource를 무작위적으로 선택한다 가정하였다. 선정한 Capacity는 아래와 같다.

Terminal Name	A			B		C			D			E			
TSA Gate	A12	A21	A35	B30	B9	C10	C21	C30	D18	D22	D30	E8	E16	E33	E33
Capacity	20	20	22	24	22	28	32	29	45	45	35	28	32	32	25

또한 출국 검사 과정의 Process 모듈의 전후로 Assign 모듈을 삽입하여 해당 과정에 소요되는 <대기시간 + 검사 소요시간>을 기록하였다. 이렇게 기록한 수치를 15분 단위로 Variable에 Assign 한 뒤 ReadWrite 모듈을 이용해 csv파일로 저장하여 Resource Capacity를 설정하기 위한 비교값으로 이용하였다. 15분 단위로 Assign하는 과정은 MOD 함수를 이용해 TNOW를 15로 나눈 나머지가 0인 경우 기록하는 과정으로 이용했다.

8) Delay(분포 가정은 아래에 상술, 터미널 별로 상이함)

→ Record(Transfer Time)

→ ReadWrite(Transfer Time, Arrival Time – TNOW, csv output)

3)번과 유사한 이유로 최종적으로 게이트까지 이동하는 과정에 대해서는 Delay 모듈을 이용하였다. 터미널 별로 각 TSA Security Check에서 출발 게이트까지의 직선거리를 구하여 평균 걷기 속도를 나누고, 일반적으로 출국 심사 과정을 자신의 출발 게이트와 가까운 곳에서 진행할 것이라는 가정을 하였다. (다른 모듈들과 다르게 유일하게 시간 단위를 Seconds로 구성하였다.)

✓ 터미널 A: TRIA(60 , 105, 150)

✓ 터미널 B: TRIA(75, 121.5, 180)

✓ 터미널 C: TRIA(60, 127,5, 188.25)

✓ 터미널 D: TRIA(75, 165, 375)

✓ 터미널 E: TRIA(56.25, 121.5, 266.25)

또한 Record 모듈을 이용해서 환승 시간을 Arrival Time를 이용한 Time Interval 수치로 환승시간에 대한 통계값을 구해보고자 하였다. 마지막으로 Dispose하는 과정에서 Decide로 Entity의 구분을 바탕으로 환승 승객을 구분해낸 뒤 Create 시뮬레이션 과정의 Delay없이 부여되어 환승승객의 공항 도착 시간을 나타내는 Arrival Time를 TNOW에서 뺀 값을 ReadWrite 모듈로 csv파일(DFW_output.csv)에 저장하여 각 승객의 환승시간을 기록했다.

5. 모델 검증 계획

위 과정들을 따라서 시뮬레이션을 실시했으며, 공항의 혼잡도의 기준으로 이용하였던 Security Wait Time의 데이터가 TSA Security Check의 closing time으로 인해 오후 10시 이후에는 존재하지 않는 경우가 있었다. 이에 더해, 각 TSA Security Check의 opening time까지 고려해 replication을 04:00 am – 10:00 pm의 18시간 동안 수행하였으며, 앞서 4.4, 4.7의 과정에서 modular 함수를 이용하기 위해 base time unit은 minutes으로 설정하였다. 또한 각 Replication을 통해서 구한 환승 시간의 분포들의 통계값의 신뢰성을 위해 30회의 Replication을 진행한 결과를 바탕으로 MCT에 대한 검증을 실시하였다.

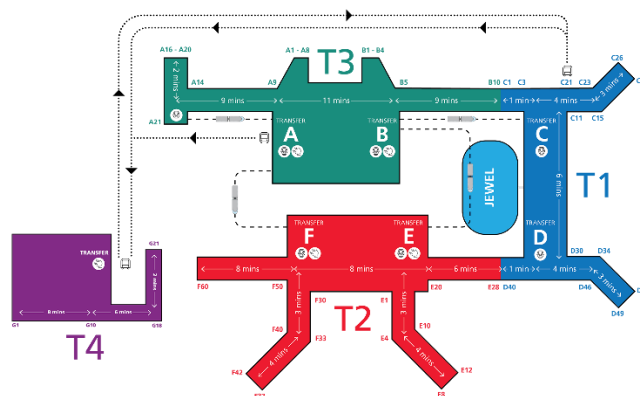
DFW 국제공항의 MCT 검증을 위한 시뮬레이션 모델은 실제 공항의 상황과 유사한 혼잡도를 구현해내고자, 실제 존재한 데이터를 이용하였다. 또한 알려진 내용이 없는 TSA Security Check 별로의 Counter의 개수를 경험적으로 시뮬레이션을 반복하여, TSA의 Wait Time이 실제 실시간 24시간 동안의 Wait Time과 유사한 양상을 띄도록 조절하였다.

또한, 항공편의 도착/출발에 따라 승객이 물리는 상황을 반영하고자 entity의 arrival을 항공기 기준으로 하여, 항공기가 도착했을 때 탑승객을 표현하는 entity가 한 번에 122개가 생성되도록 하였다.

D. 창이 국제공항 (Singapore Changi International Airport, SIN)

i. 공항설명

1. 공항구조



- 창이공항은 대표적인 환승 허브 공항으로써 T1, T2, T3, T4 총 4개의 터미널로 구성되어 있다. 싱가포르의 도시국가인 관계로 국내선이 존재하지 않고 모든 항공편을 국제선으로 운항하며, 환승 또한 국제선과 국제선 간의 환승만이 존재한다.

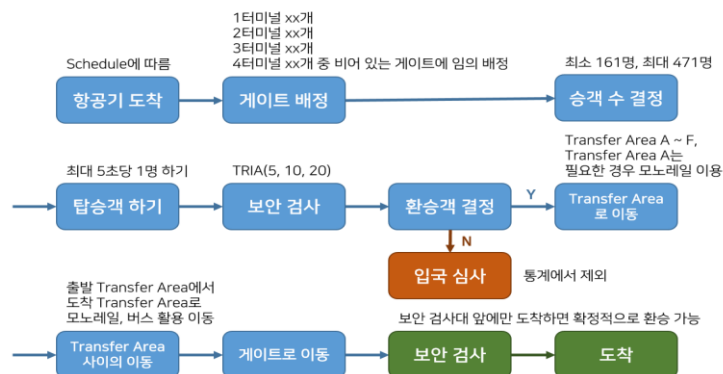
- 탑승 수속을 거치지 않은 상태에서 이용할 수 있는 Public Area에서는 각 터미널을 도보로 오갈 수 있지만, 환승객들이 위치한 Secured Area 안에서의 터미널 간 이동을 위해서는 반드시 모노레일(1, 2, 3터미널 간 이동)이나 버스(4터미널 발착의 이동)를 이용해야 한다.
- 1, 2, 3터미널은 각 2개씩의 Transfer Area를 가지고 있는데, 1터미널은 C, D, 2터미널은 E, F, 3터미널은 A, B Transfer Area로 정의된다. 환승객은 자신이 하기한 게이트에서 Transfer Area로 이동한 후, 해당 Transfer Area에서 다음 탑승 게이트가 속하는 Transfer Area로 이동을 완료한 다음 게이트를 찾아간다. A Transfer Area만 유일하게 Transfer Area 내부를 이동할 수 있는 모노레일이 설치되어 있다. 나머지 Transfer Area 내에서의 모든 이동은 도보로 이루어진다.
- A-F(2-3터미널), B-C(1-3터미널), B-E(2-3터미널) 간에 모노레일이 설치되어 있고, 터미널 간 이동은 모노레일을 통해서만 이루어진다. 1터미널과 2터미널 사이에서 이동하기 위해서는 반드시 모노레일을 두 번 이용하여야 하며, 이외의 경우는 모두 한 번의 모노레일 이용이 필요하다.
- 4터미널로 이동하기 위해서는 버스를 이용하여야 하며, 버스는 A Transfer Area와 C Transfer Area에서 출발한다. C-A-4의 순환 구조를 가지고 있으며, 배차 간격이 모노레일에 비해 길고 탑승 정원이 크지 않기 때문에 한 번 버스를 놓치는 것만으로도 심각한 환승 지연을 유발할 수 있다.

2. 특이사항

- 미국의 공항들과 다르게 환승객들은 Public Area로 나가서 재발권 절차를 거칠 필요 없이 Secured Area 안에서 2번의 보안 검사(도착 게이트를 지날 때 한 번, 출발 게이트에 도착하기 직전 한 번)와 다음 탑승 게이트로의 이동만을 완료하면 되기 때문에 모노레일과 버스를 반드시 사용한다고 볼 수 있다.

ii. 모델설명

1. 전체 구조



- 모델의 전체 구조는 위와 같다. [1] Schedule에 따라서 항공기가 공항에 도착한다. [2] 이후 비어있는 게이트에 임의로 배정되고, [3] 항공기에 타고 있는 승객 수가 결정된다. [4] 해당 게이트에서 탑승객이 하기하는데, 게이트 여러 개가 하나로 묶여 있기 때문에 해당 그룹에 주기되어 있는 항공기의 수가 많을수록 하기 속도가 빨라진다. [5] 이후 게이트 바로 앞에서 모든 승객이 보안 검사를 거치고 [6] 정해진 비율에 따라 환승객과 환승하지 않는 승객이 구분되어 [6.5] 환승하지 않는 경우에는 입국 심사를 받기 위해 다른 줄로 이동하므로 통계에서 제외된다. [7] 환승객은 자신의 게이트에 해당하는 Transfer Area를 향해 도보로 이동한 후(Transfer Area A에 한해서는 필요한 경우 모노레일을 이용할 수 있다) [8] 자신이 다음으로 탑승할 비행기가 위치한 Transfer Area로 이동한다. 이때는 모노레일과 버스를 활용하여 이동한다. 이동이 완료된 후에는 [9] 도보 이동(Transfer Area A에 한해 모노레일 이용 가능)을 통해 최종적으로 환승할 게이트에 도착한다. 보안 검사대 앞에만 도착하면 확정적으로 환승이 가능한 구조이므로 보안 검사에 걸리는 시간은 환승 소요 시간에서 제외되었다.

2. 구조 가정

- 창이 공항은 여러 개(일반적으로 2~3개, 많게는 8개)의 게이트가 그룹으로 묶여서 하나의 보안 검색대를 공유한다는 특징이 있다. 이러한 특징에 근거하여 같은 보안 검색대를 사용하는 게이트를 묶어서 나타내었으며, 최대 해당 그룹에 속한 게이트의 수만큼 항공기가 배정받을 수 있도록 구현하였다. 예를 들어, 5개의 게이트를 묶은 그룹은 동시에 최대 5개의 항공기가 배정될 수 있다.

- 실제 공항에서 발생하는 절차와 같이 항공기가 활주로를 통해 착륙하면 비어 있는 게이트 중 하나를 배정받는 식으로 모델을 구성하였다. 이미 승객이 하기하고 있는 게이트에는 항공기가 새롭게 주기할 수 없으므로, 승객이 내리고 있는 게이트에 배정되었다면 확인 후 다른 게이트에 다시 배정될 수 있도록 구현하였다.

- 국제선 도착 항공편은 모든 게이트에 걸쳐 고르게 도착한다고 가정하였다.

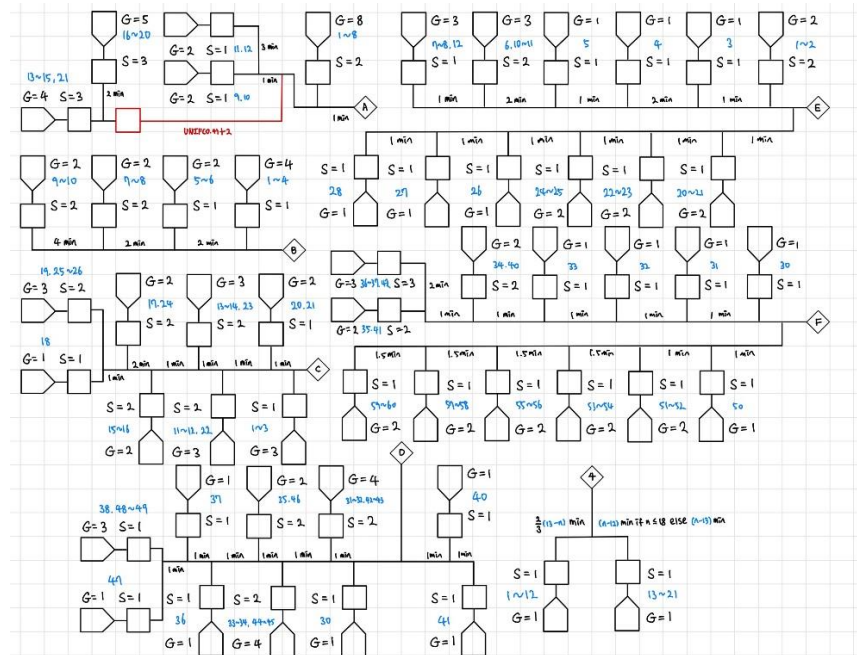
- 항공사에 따른 차이는 고려하지 않는다.

3. 데이터 가정

- 창이 공항의 공식적인 2019년 통계¹⁴에 따르면, 월 평균 31,800대의 민항기가 착륙하며 이는 곧 하루에 약 1060대의 민항기가 승객 수송을 위

¹⁴ Changi Airport Air Traffic Statistics, <https://www.changiairport.com/corporate/our-expertise/air-hub/traffic-statistics.html>, 2023.12.18.

해 착륙한다는 것을 의미한다. 1시간 동안 착륙하는 비행기가 하루 동안 착륙하는 전체 비행기에 대해 얼마의 비율을 차지하는 지에 관련된 데이터가 존재하므로, 해당 데이터를 통해 Schedule을 구성하고 Scale Factor를 활용하여 하루 동안 평균적으로 1060대 근처로 착륙할 수 있도록 데이터를 설정하였다.



- 게이트를 균등하게 분포하기로 가정하였으므로, 탑승객이 타야 하는 다음 비행기 또한 균등하게 분포되어 있어야 한다. 따라서 각 Transfer Area로 이동할 확률은 (해당 Transfer Area에 속한 게이트의 개수) / (전체 게이트의 개수)로 균등하게 비례하도록 설정하였으며, Transfer Area에 도착한 이후 게이트로 이동할 때도 (그룹 내에 속한 게이트의 개수) / (해당 Transfer Area에 속한 게이트의 개수)를 통해 계산하였다.

- 비행기에 탑승하고 있는 사람의 수는 창이 공항을 허브 공항으로 사용하는 싱가포르항공(Singapore Airlines)의 항공기 자료를 바탕으로 최소 161명에서 최대 471명 사이의 정수로 결정된다.

- 비행기 지연은 창이 공항 공식 사이트를 기준으로 15분 이상 지연되는 항공기가 약 15%였으므로 중앙값이 10분이고 표준편차가 5분인 분포를 설정하여 15분을 초과하는 비행기가 16% 정도가 되도록 설정하였다. 또한, 음의 방향으로 과도한 이상치를 제거하기 위하여 아무리 빨라도 원래의 도착 시간보다 5분 이상은 빨리 도착하지 못하도록 설정하였다.

- 승객이 하기하는 시간은 비행기가 해당 게이트 그룹에 n대 주기하여 있는 경우 $5 / n$ 초가 걸리도록 설정하였다. 이는 해당 게이트 그룹에 주기되어 있는 항공기의 수가 많을수록 하기 속도를 빠르게 만들기 위함이다.

- 보안 검사에 소요되는 시간은 보안 검사대(금속 탐지 기계)를 통과하는

데에 걸리는 시간으로 충분하므로 초 단위로 TRIA(5, 10, 20)으로 가정하였다. 짐을 내려놓거나 찾는 것은 Queue 안에서 이루어지는 것이 아니라 보안 검사대를 통과하기 전과 후 줄 밖에서 이루어지는 행위이기 때문에 시간에 산입하지 않았다. 각 게이트 그룹의 보안 검사대 수는 창이 공항의 공식 홈페이지 지도¹⁵를 참고하였다.

- 모든 승객의 약 25%가 환승을 하는 것으로 가정하였다.
- 도보로 걸리는 시간과 모노레일, 버스로 소요되는 시간은 창이 공항의 공식 홈페이지를 참고하였다. 모노레일은 일반적으로 4분에서 6분 사이의 간격으로 도착하고, 이동하는 데에는 그의 절반인 2분에서 3분 정도가 소요된다. 버스의 경우에는 일반적으로 12분 간격으로 도착하고 이동하는 데에는 각 터미널 별로 12분 또는 18분이 소요된다.
- 모노레일과 버스는 항상 매 분 정각에 출발하는 것을 기준으로 삼았으며, 4분 단위로 도착하는 모노레일은 한 쪽에서는 매 4n분마다 출발하고, 반대쪽에서는 매 4n+2분마다 출발한다. 6분 단위로 도착하는 모노레일은 한 쪽에서는 매 6n분마다 출발하고, 반대쪽에서는 매 6n+3분마다 출발한다. 버스는 Transfer Area C를 기준으로 매 9n+3분마다 도착하고, Transfer Area A를 기준으로 매 9n분마다 도착한다. 이는 C에서 A까지 버스로 이동하는 시간이 6분 걸리는 것을 구현하기 위함이다.
- 달리는 사람이나 천천히 이동하는 사람을 고려하여 모든 이동 과정에는 TRIA(-0.5, 0, 0.5)의 무작위성을 부여하였다. 이는 모노레일이나 버스와 같이 정시 운행하는 교통수단에는 적용되지 않으며, 오로지 Route로 정의된 도보 이동에만 적용된다.

4. 모델 세부 설명

1) Airplane Entity

- Airplane Entity의 생성과 게이트 배정, 승객 수 결정을 통해서 어떤 게이트에서 얼마나 많은 승객이 나오는지가 결정된다.
- i. Airplane Arrives (Create)
 - 정해진 Schedule에 따라서 Entity 'Airplane'이 Create된다.
- ii. Arrival Selection (Decide)
 - Decide를 통해 항공기가 무작위로 게이트에 배정된다. 이때 각 게이트 그룹별로 배정될 확률은 동일하다.
- iii. Occupancy Checker (Decide)
 - 해당 게이트 그룹에 이미 최대 수의 항공기가 주기되어 있다면 더 이상 배치가 불가능하므로 몇 대의 항공기가 이미 들어와

¹⁵ Changi Airport Maps, <https://www.changiairport.com/en/maps.html>, 2023.12.18.

있는지를 알려주는 Variable인 'XNN Airplane'과 게이트 그룹의 항공기 최댓값을 비교하여 XNN Airplane이 최댓값과 동일하면 다시 (ii)의 결정 단계로 되돌려보낸다.

- X는 Transfer Area, NN은 게이트 번호를 의미한다.

iv. Gate XNN (Assign)

- 여러 개의 게이트가 묶여서 그룹이 형성된 경우 게이트 번호가 여러 개 적혀 있는 경우 또한 존재한다.

(예를 들어, 'A16 to 20'이나 'A13, 14, 15, 21'이 있다.)

- 게이트 배정이 완료되면 'XNN Airplane'에 1을 더한다.

- 해당 게이트에서 하기할 수 있는 승객의 최댓값인 'XNN MAX'에 Airplane Size를 더한다. Airplane Size는 앞선 데이터 가정과 같이 최소 161명에서 최대 471명 사이의 정수로 결정되는, 항공기에 탑승 중인 사람의 수를 의미한다.

v. Gate X Assigned (Dispose)

- Entity 'Airplane'은 Assign 작업이 모두 끝나면 Dispose된다. 이때 Transfer Area별로 Dispose 모듈을 나눈 이유는 각 Area에 몇 대의 항공기가 배정되었는지 관측하기 쉽도록 하기 위함이다.

2) Traveler Entity

- Traveler Entity가 이동하는 과정에서 걸린 시간을 측정하면 각 승객이 환승에 걸린 시간을 시뮬레이션적으로 파악할 수 있다.

i. XNN (Create)

- Entity 'Traveler'가 5 / XNN Airplane초마다 생성된다. 이는 곧 해당 게이트 그룹에 항공기가 한 대 있으면 탑승객이 5초마다 한 명, 두 대 있으면 2.5초마다 한 명...과 같이 하기함을 의미한다.

ii. Real Arrived Traveler XNN (Decide)

- Arena 프로그램의 한계 상 Create 모듈에 직접적으로 MAX를 설정하면 그 값이 Variable이 아닌 Constant로 인식되어 그 값이 추후에 업데이트가 되더라도 Create 모듈에는 반영되지 않음을 발견했다. 따라서 (i)의 Create 모듈은 Entity를 제한 없이 생성하되, 이 모듈에서 현재까지 이 게이트 그룹에서 하기한 사람의 수인 XNN Now와 XNN MAX를 비교하여 XNN Now가 XNN MAX보다 작은 경우에만 Entity가 다음 단계로 넘어갈 수 있도록 한다.

iii. Assign XNN (Assign)

- XNN Now가 XNN MAX보다 작아서(아직 내릴 승객이 남아 있어서) Entity가 다음 단계로 진행한 경우 아래의 과정을 진행한다.

- Entity 한 개가 더 진입하였으므로 가장 먼저 XNN Now에 1을 더한다.
 - 이후 XNN Now와 XNN Max가 동일해지면(하기해야 하는 마지막 승객이었다면) 비행기가 더 이상 주기해 있을 필요가 없으므로 XNN Airplane을 0으로 바꾼다.
 - Entity의 Attribute인 Arrival Time에 TNOW(현재 시간) 값에서 비행기 지연으로 인해 추가된 시간만큼을 뺀다.
- iv. XNN Security (Process)
- 모든 Entity는 환승 여부에 상관없이 게이트에서 나오자마자 보안 검사를 거치게 된다. 보안 검색대(Resource)의 개수와 소요 시간인 Security Check Time은 데이터 가정에서 정한 값을 따른다.
- v. Transfer Decision for XNN (Decide)
- 미리 정한 환승객 비율에 따라 환승하는 사람과 환승하지 않는 사람을 구분한다. 환승하지 않는 사람은 통계적으로 고려 대상이 아니므로 제외하고, 환승객에 해당하는 Entity만 다음 단계로 진행한다.
- vi. To Transfer X (Route)
- 자신이 내린 게이트가 속한 Transfer Area로 이동하기 위해 Route를 활용하여 여러 경로를 거친다. 이때, 멀리 위치한 게이트 그룹의 경우 다른 게이트 그룹이 거친 경로를 동일하게 지나는 경우도 있다. 또한, Transfer Area A 중에서 A13부터 A21의 게이트는 Shuttle Train(모노레일)을 이용하여 이동하게 된다.
- 3) Shuttle Train(모노레일)과 Shuttle Bus(버스)
- i. Shuttle Train, Shuttle Bus (Station)
- Transfer A Shuttle Train
 - Shuttle Train Transfer A to F, F to A, B to E, E to B, B to C, C to B
- 위와 같이 총 7가지의 Shuttle Train Station이 존재한다.
- Shuttle Bus A to 4, Shuttle Bus C to 4
- 총 2가지의 Shuttle Bus Station이 존재한다. Bus 또한 Train과 같은 과정으로 진행된다.
- Terminal 4에서 출발하는 인원은 계산의 복잡성을 이유로 제외되었기 때문에 Terminal 4에 도착하는 인원만이 통계에 포함된다.
- ii. Assign Transfer A, Assign Train X to Y, Assign Bus X to 4 (Assign)
- 해당 Station에서 다음 열차가 올 때까지 대기해야 하는 시간

에 대한 Attribute인 WaitingX에 $(\text{최대 대기시간}) - (\text{현재 시간} \bmod \text{최대 대기시간})$ 의 값을 더한다. 반대편 승강장의 경우에는 $(\text{최대 대기시간}) - ((\text{현재 시간} + 0.5 * \text{최대 대기시간}) \bmod \text{최대 대기시간})$ 을 더한다.

- iii. Transfer A Shuttle Train Wait Decision, Shuttle Train X to Y, Shuttle Bus X to 4(Decide)
 - 현재까지 열차에 탑승한 승객이 해당 열차의 정원을 초과하면 나머지는 Use Next라는 새로운 줄으로 보내서 이번 열차가 떠난 후 다음 열차에 탑승하도록 한다. 각 열차의 정원은 Transfer A Shuttle Train MAX 또는 Shuttle Train X to Y MAX이라는 Variable로 결정되어 있다.
 - Transfer A Shuttle Train MAX = 50
 - Shuttle Train A to F MAX = 80
 - Shuttle Train B to E MAX = 100
 - Shuttle Train B to C MAX = 70
 - Shuttle Bus A to 4 MAX = 100
- iv. Transfer A Shuttle Train Waiting Line, Shuttle Train X to Y Waiting Line, Shuttle Bus X to 4 Waiting Line (Delay)
 - 열차의 정원이 초과하지 않았을 때 열차에 성공적으로 탑승할 수 있는 사람들은 Assign에서 결정된 WaitingX의 값만큼 대기한 이후 열차를 통해 목적지로 이동한다.
- v. Transfer A Shuttle Train Use Next, Shuttle Train X to Y Use Next, Shuttle Bus X to 4 Use Next (Delay)
 - 열차의 정원을 초과한 경우 Use Next라는 새로운 줄에서 대기한 이후 다시 (ii)의 Assign으로 이동하여 새로운 WaitingX 값을 부여 받고 다음 열차의 줄에 서게 된다.
- vi. To Transfer A After Shuttle (Route)
 - Transfer Area A 내에서는 모든 승객의 목적지가 Transfer Area A로 동일하므로 Decision 없이 바로 Transfer A를 향한 길로 이동한다.
 - 이는 Terminal 4로 향하는 Bus에서의 과정과 동일하다.
- vii. Shuttle Train X to Y Decision (Decide)
 - Transfer Area 간의 이동을 담당하는 Shuttle Train은 승객마다 목적지가 다양하므로 Decide를 통해 각 승객의 목적지별로 이후 이동 경로를 분리한다.
- viii. X to Y to Z (Route)

- 승객의 다음 탑승 게이트가 위치한 Transfer Area로 각각 이동한다. 각 이동 시간은 데이터 가정에서 결정한 시간을 따른다.

4) Transfer Area to Transfer Area

- i. Transfer X (Station)
 - 해당 Transfer Area에서 하기한 승객은 모두 한 곳으로 모여서 각자의 목적지로 이동하게 된다.
- ii. Departure X (Decide)
 - 승객이 이동할 다음 Transfer Area를 데이터 가정에서와 같이 각 Transfer Area에 포함되어 있는 게이트의 수와 균등 비례하도록 배분한다.
- iii. Assign X to Y (Assign)
 - Arrival이라는 새로운 Attribute에 목적지 Transfer Area를 string 값으로 추가한다.
- iv. X to Y (Route)
 - 다음 Transfer Area로 Route를 통해 이동한다. 도보로 이동할 수 있는 경우에는 바로 Arrival Y라는 최종 목적지 Station에 도착하고, 교통수단을 이용하여야 하는 경우는 C에서 정의한 모듈을 각 경로마다 적절히 거쳐 이동한다.
- v. Arrival Y (Station)
 - 승객이 최종 목적지가 있는 Transfer Area에 도착하면 목적지 게이트에 따라 다른 소요 시간을 배정한다.
- vi. Assign Arrival Y (Assign)
 - Gate라는 Attribute에 (각 게이트 그룹 별로 속한 게이트 수) / (해당 Transfer Area에 속한 총 게이트 수)만큼의 확률으로 게이트 그룹을 균등하게 배정한다. 해당 게이트 그룹까지 소요되는 시간은 B의 (vi)에서 이미 계산이 완료되었으므로, 해당 값을 그대로 대입한다.
- vii. Gate Decision A, Gate Decision 4 (Decide)
 - Transfer Area A는 Shuttle Train을 탑승해야 하는 경우가 있으므로 해당 경우를 따로 처리한다.
 - Transfer Area 4는 구조상 게이트 번호를 3그룹으로 묶어서 표현할 수 있으므로, 해당 경우를 각각 처리한다.
- viii. Gate Y (Route)
 - 최종 통계 계산과 Dispose를 위해 Route를 이용하여 이동한다. 이때 데이터 가정에 언급한 무작위성이 추가된다.
 - 여기까지 도착하면 더 이상 추가적으로 소요되는 시간은 없다

고 가정한다. (보안 검색대 바로 앞)

- 5) Final
 - i. Final (Station)
 - 최종 통계 계산과 Dispose를 위해 모든 환승객 Entity가 한 곳으로 모인다.
 - ii. Time Record (Record)
 - TNOW와 Arrival Time을 비교하여 Time Interval이라는 Tally에 추가한다.
 - iii. MCT Exceeded (Decide)
 - TNOW – Arrival Time이 기존에 공항에서 설정한 MCT를 초과한 경우 ReadWrite Y, 아닌 경우 N으로 보낸다.
 - iv. ReadWrite Y, ReadWrite N (ReadWrite)
 - ReadWrite에서는 csv 파일에 한 개의 Transferring Traveler Entity 당 Replication 번호(NREP), Transfer Time(Time Interval과 동일한 값), MCT Exceeded(Yes or No)을 추가한다. 이때, ReadWrite Y에서는 MCT Exceeded의 값이 2, ReadWrite N에서는 값이 1이다.
 - v. Transferring Traveler (Dispose)
 - 최종적으로 모든 Entity가 Dispose되면서 시뮬레이션이 종료된다.
5. 모델 검증 계획
 - 총 30번의 replication이 돌아가는 동안 하루 간 도착하는 항공기 대수의 평균이 실제 하루동안 도착하는 국제선 항공편 수인 1060편과 유사하면 모델이 검증된 것으로 판단할 것이다.
 - 구체적으로, 항공기 대수가 $[1060 * 0.95, 1060 * 1.05]$ 의 구간 내에 있는 경우 모델이 실제와 유사하게 설계되었다고 판단할 것이다.

E. 인천 국제 공항(Incheon International Airport (Seoul), ICN)

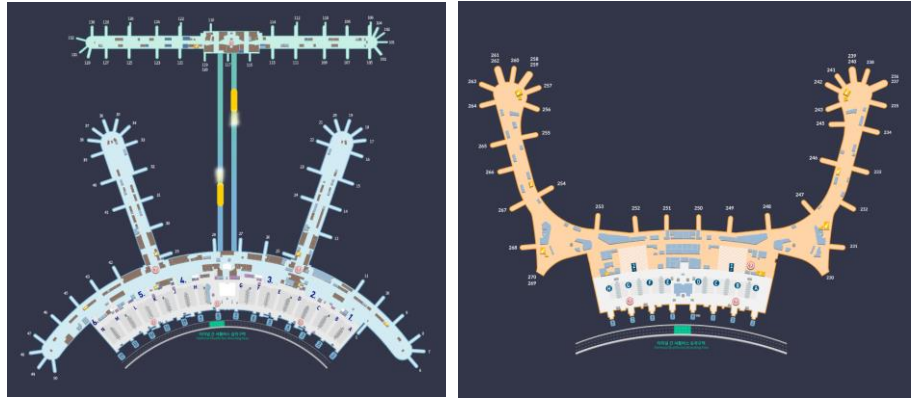
iii. 공항설명

지난 2001년 동북아 허브공항을 목표로 개항한 인천국제공항은 국민들의 성원, 정부의 비전과 정책, 7만여 공항 종사자들의 헌신적인 노력에 힘입어 국제화물 2위, 국제여객 5위의 허브공항이자 세계의 하늘길을 잇는 핵심 거점으로 도약할 수 있었다.

대한민국 최대 규모의 민간 공항이자, 북한에 인천공항보다 규모가 큰 공항이 없으므로 자연스럽게 한반도 최대 규모의 공항이기도 하며, 해외 국가와 한국 간 왕래가 이루어질 때 대부분의 국제선 여객기들이 이착륙하는 곳이기에 대

한민국의 하늘길 대문으로 불리우고 있다.

1. 공항구조



- 인천국제공항(ICN)은 총 2개의 터미널(제1여객터미널, 제2여객터미널)로 이루어져 있으며 제1여객터미널은 탑승동(Concourse)을 포함하고 있다. 각 터미널은 모두 각기 다른 모양이며, 좌우가 대칭을 이루는 것이 특징이다. 입/출국 게이트 수나 출국을 위한 보안검사대의 숫자 등에 차이가 존재한다.
- 터미널 간의 이동은 교통 수단을 통해 이루어지며, 교통 수단으로는 Secured Area 내에서는 셔틀트레인을 이용할 수 있고, 밖에서는 공항 셔틀버스, 공항철도 등을 이용할 수 있다.
- 제1여객터미널에는 총 1-50(4, 13, 44 게이트 제외)까지 47개의 게이트가 있고, 탑승동(Concourse)에는 101-132게이트(116,120 제외)까지 총 30개의 게이트가 있다. 그리고 제2여객터미널에는 231-270까지 총 40개의 탑승게이트가 존재한다.
- 각 터미널마다 환승을 위한 보안검색장과 티켓을 발급할 수 있는 환승데스크가 마련되어 있다. 제1여객터미널에는 2곳의 보안검색장과 4곳의 환승데스크가 존재하지만, 2층에 위치한 2개의 환승데스크는 보안검색 전에 사용 가능한 데스크이기에 보안 검색 이후 티켓 발급이 가능한 4층의 환승데스크만을 고려하였다. 탑승동에는 2곳의 보안검색장과 1곳의 환승데스크가 있다. 마지막으로, 제2여객터미널에도 역시 2곳의 보안검색장과 4개의 환승데스크가 마련되어 있지만 보안 검색 후에 이용 가능한 2개의 환승데스크만을 고려하였다. 각각의 보안 검색장에는 6대의 보안검색대가 설치되어 있고, 환승데스크에는 4개의 창구가 있다.

2. 특이사항

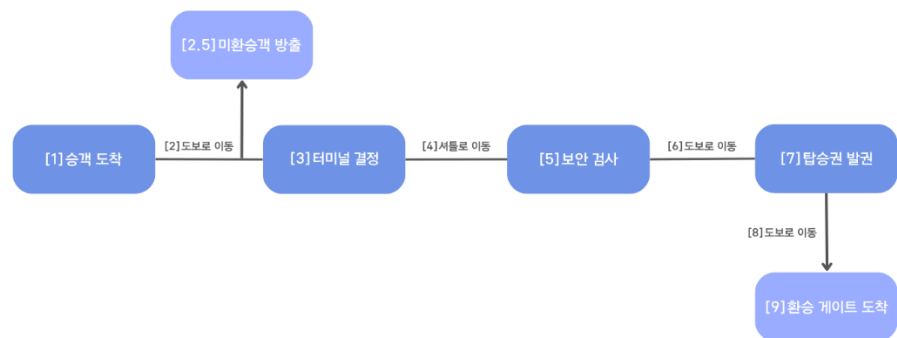
- 공항의 구조가 탑승동을 기준으로 양쪽으로 제1여객터미널과 제2여객

터미널이 위치하고 있는 형태이며, 제1여객터미널과 탑승동, 그리고 탑승동과 제2여객터미널 사이에 셔틀트레인이 지나는 경로가 마련되어 있어 제1여객터미널에서 제2여객터미널로의 이동을 위해서는 탑승동에서 셔틀트레인 환승을 거쳐야 한다.

- 어느 터미널에 도착하고, 어느 터미널로 이동하여 환승하는지에 따라 어떤 보안검색대와 환승데스크를 이용할 것인지가 결정되기 때문에 환승 과정에 있어서 다양한 경로가 발생한다.

iv. 모델설명

1. 전체 구조



- 모델의 전체적인 구조는 다음과 같다. [1] 각 게이트를 통해 인천공항으로 입국한 모든 승객(환승객+미환승객)이 도착한다. [2] 모든 승객들은 보안 검색대까지 도보로 이동하며, [2.5] 미환승객용 입국 심사대와 환승객용 보안 검색대가 나뉘는 장소에서 미환승객이 방출된다. [3] 환승객은 어느 터미널에서 환승을 진행할 것인지에 따라 보안검사를 먼저 받을지, 터미널을 이동한 후 보안검사를 받을지를 결정하게 되고, [4] 다른 터미널에서 환승이 이루어지는 승객들은 셔틀트레인을 타고 터미널을 이동하게 된다. [5] 터미널을 이동한 환승객과 이동하지 않은 환승객 모두 보안검사를 받게 되며, [6] 보안검사를 마치면 각 터미널에 위치한 환승데스크까지 도보로 이동한다. [7] 그리고 그곳에서 탑승권을 발급받고, [8] 각 환승객들이 할당받은 환승 게이트로 이동하여 기내 탑승 및 환승을 마치게 된다.

1) 승객 도착

인천공항에 도착하는 모든 승객들은 각 터미널의 2층에 위치한 입국 게이트를 통해 공항 내부로 진입하게 된다. 환승객과 환승을 하지 않는 일반 승객이 같은 경로로 입국장을 빠져나오며, 각 게이트의 위치에 따라 승객이 다음 행선지까지 도보로 이동하는 시간이 결정된다. 이러한 이유로 실제 120개에 달하는 게이트를 모델링하기 위해 공항 내 위치에 따라 거리가 인접한 약 3-9개 정도의 게이트를 그룹화하여

모델을 구현했다.

2) 보안 검색장까지 이동

각 터미널에 위치한 도착 게이트에서 빠져나온 승객들은 같은 경로로 입국 심사대 및 보안 검색장으로 이동하게 된다. 환승객과 미환승객이 섞여서 이동하다가 목적지에 다다르면 미환승객은 방출된다. 게이트마다 보안 검색장까지의 거리가 달라 도보 소요시간 역시 차이가 발생한다. 또한, 어느 터미널의 환승 게이트에서 출국이 진행되는지에 따라 보안검색이 먼저 진행될지, 터미널 이동이 먼저 진행될지가 나뉘게 된다.

3) 터미널 결정

승객이 도착한 터미널을 기준으로 다른 터미널에서 출발이 진행된다면, 그 승객은 셔틀트레인을 이용하여 터미널을 이동한다.

다른 터미널로 이동하는 경우는

- 기존에 도착한 터미널에서 보안 검사를 실시하고 셔틀트레인을 이용하여 환승 터미널로 이동하는 경우

- 기존에 도착한 터미널에서 셔틀트레인을 이용하여 다른 터미널로 이동한 후 이동한 터미널에서 보안검사를 실시하는 경우

의 두 가지 경우로 나뉜다. 본 시뮬레이션에서는 승객의 환승 경로에 따라 다른 구조를 설정하여 두 가지 경우를 모두 고려하였다.

4) 셔틀로 이동

터미널 간의 이동은 제1여객터미널과 제2여객터미널 사이에 위치한 탑승동으로부터 양쪽 쌍행선으로 설치되어 있는 셔틀트레인 경로를 따라 진행되는데, 탑승동과 나머지 두 터미널 사이의 이동은 한 번의 탑승으로 가능하지만, 제1여객터미널과 제2여객터미널 간의 이동은 탑승동에서의 환승을 거쳐 2번의 탑승이 이루어져야 가능하다.

5) 보안 검사

3)에서 설명한 것처럼 두 경우 모두 결국 보안 검사를 실시하게 되는데 보안 검사는 각 터미널의 중앙에 위치한 2곳의 보안검색장에서 진행된다. 모든 환승객들이 2곳으로 몰리게 되므로 Queue의 과도한 축적을 방지하기 위해 보안 검색에 소요되는 시간을 적절하게 조정하였다.

6) 환승데스크까지 도보로 이동

보안 검사가 진행된 후 출발을 위해 환승데스크에서 탑승권을 발권해야 한다. 제1여객터미널과 제2여객터미널은 2층에서 보안검색이 진행된 후 4층에 위치한 환승데스크에서 티켓을 발권할 수 있고, 탑승동은 2층에서 보안검색 진행 후 3층 환승데스크에서 발권이 가능하다. 이와

같이 보안 검색대와 환승데스크의 위치는 터미널마다 다르기 때문에 각각의 경로에 따른 이동 시간을 개별적으로 고려하여 가정하였다.

7) 탑승 티켓 발권

환승데스크는 제1여객터미널과 제2여객터미널에 2곳, 그리고 탑승동에 1곳이 마련되어 있고, 각 데스크마다 4명의 직원이 발권을 돕는다. 각 터미널에서 환승을 준비하는 모든 환승객들이 환승데스크로 몰리기 때문에 이때에도 Queue가 발생한다.

8) 환승 게이트까지 이동

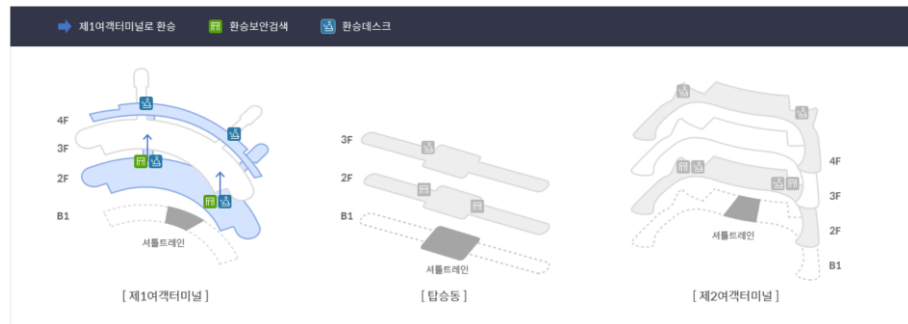
동일한 환승데스크에서 각 게이트로 이동하는 과정은 앞서 설명했던 도착게이트에서 보안 검색장까지의 이동 소요시간과 같을 것이기 때문에 이를 고려하여 예상 소요시간을 설정하였다.

9) 환승 게이트 도착

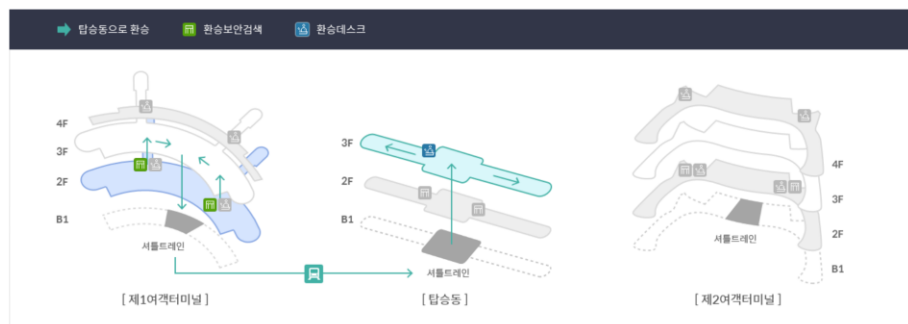
각 게이트에 도착한 승객들이 기내에 탑승하면서 환승이 마무리된다.

2. 구조 가정

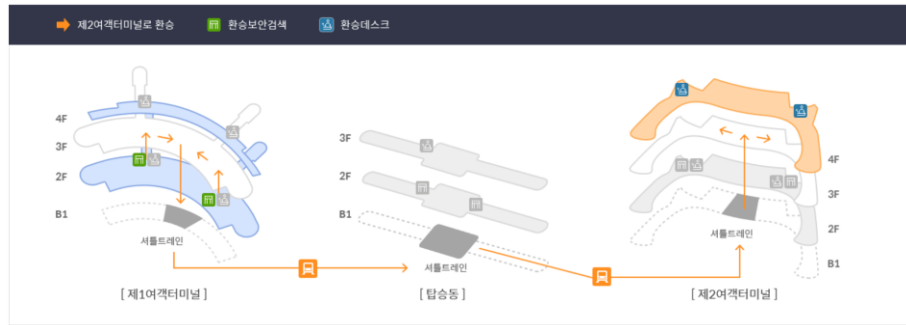
1) Terminal 1



- Terminal 1 -> Terminal 1: 도착 게이트 -> 보안 검색장 -> 환승데스크 (티켓) -> 출발 게이트

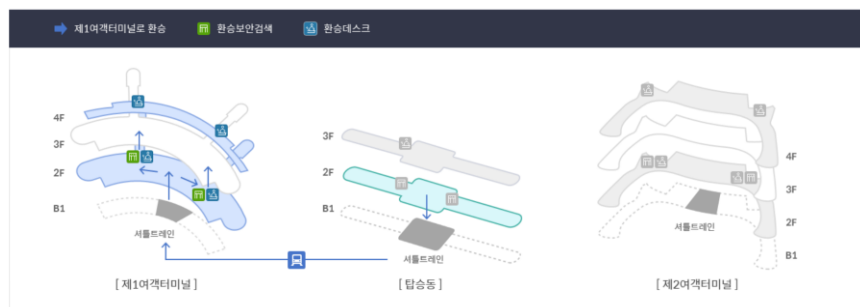


- Terminal 1 -> Concourse: 도착 게이트 -> 보안 검색장 -> 서울트레인 -> Concourse 도착 -> 환승데스크(티켓) -> 출발 게이트

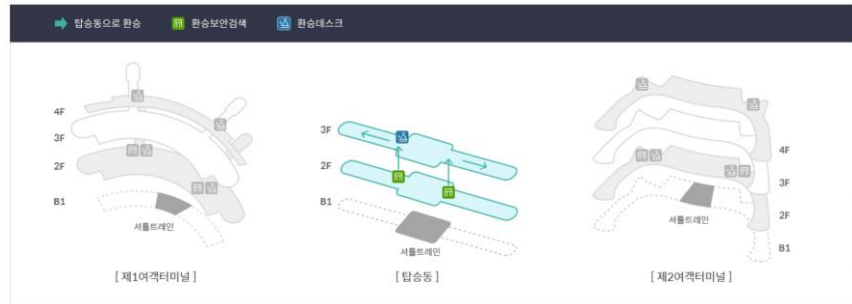


- Terminal 1 -> Terminal 2: 도착 게이트 -> 보안 검색장 -> 셔틀트레인 -> Concourse 도착 -> 셔틀 트레인 환승 -> Terminal 2도착 -> 환승 데스크(티켓) -> 출발 게이트
- 제1여객터미널의 환승 보안검색은 두 곳에서 진행되며, 왼쪽에 위치한 게이트의 승객은 보안검색1, 오른쪽 게이트의 승객은 보안검색2에서 주로 보안검색을 받는다. 한 보안검색 장소에 위치한 보안 검색대의 개수는 6대이다. 제1여객터미널에 도착하는 모든 환승객들은 제1여객터미널의 환승 보안검색대를 거친 후 이동하기 때문에 다른 두 터미널에서의 모델과는 다른 구조로 모델링하였다. 티켓을 발권하는 환승 데스크는 이동한 각 터미널에서 진행하기 때문에 제1여객터미널에 도착한 승객들은 곧바로 두 그룹으로 나뉘어 Security를 통과한 후 환승을 진행할 터미널로 이동한 다음 그곳에서 Ticketing Desk를 이용하도록 구성하였다. 제1여객터미널의 환승데스크는 두 곳이고 총 4개의 창구로 구성되어 있다. 티켓을 발급받은 승객들은 각자 출발을 위해 탑승할 게이트로 이동한다. 전반의 과정에서 각 장소 간의 승객 이동 소요시간을 각 게이트마다 고려하여 계산하였다.

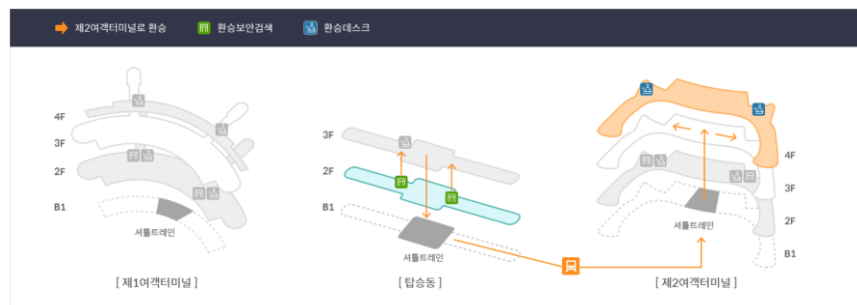
2) Concourse



- Concourse -> Terminal 1: 도착게이트 -> 셔틀트레인 -> 제1여객터미널 도착 -> 보안 검색장 -> 환승데스크(티켓) -> 출발 게이트

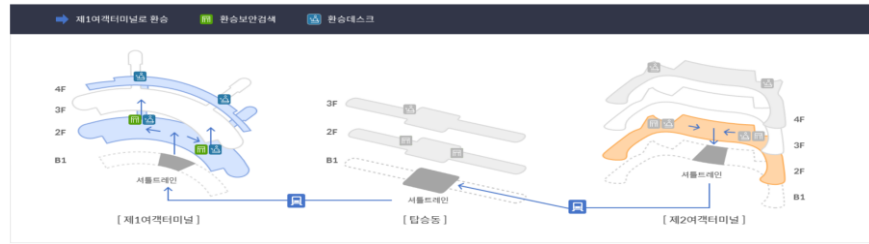


- Concourse -> Concourse: 도착 게이트 -> 보안 검색장 -> 환승데스크 (티켓) -> 출발 게이트

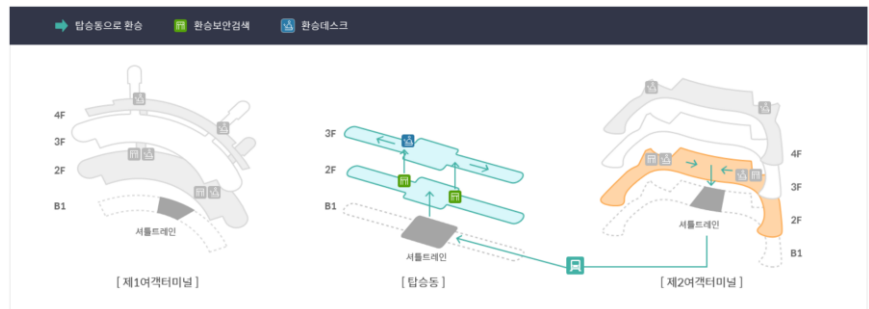


- Concourse -> Terminal 2: 도착 게이트 -> 보안 검색장 -> 셔틀트레인 -> 제2여객터미널 도착 -> 환승데스크(티켓) -> 출발 게이트
- 탑승동에 도착한 환승객 중 제1여객터미널에서 탑승하는 승객은 보안 검사를 하지 않고 바로 셔틀을 타고 터미널 이동을 진행하는 반면, 탑승동과 제2여객터미널에서 탑승하는 승객은 탑승동에서 보안 검사를 진행한 후 각 터미널로 이동한다. 탑승동의 보안 검색장은 두 곳이며 총 6개의 보안검색대를 이용할 수 있고, 탑승동에 도착한 환승객 중 제1여객터미널로 이동하는 승객을 제외한 모든 승객들이 이 두 곳의 보안검색 장소를 이용한다. 제1여객터미널에서 탑승하는 승객들은 터미널 이동을 먼저 진행하고 제1여객터미널에서 Security를 통과한 후 그곳에서 Ticketing Desk를 이용하도록 구성하였고, 탑승동과 제2여객터미널에서 탑승하는 승객들은 탑승동에서 Security를 통과한 후 각자의 터미널로 이동한 후 그곳에서 Ticketing Desk를 이용하도록 구성하였다. 탑승동과 제2여객터미널의 환승데스크는 각각 1곳과 2곳이며, 각 환승데스크에는 4개의 창구가 마련되어 있다. 티켓을 발급받은 승객들은 각자 출발을 위해 탑승할 게이트로 이동한다. 전반의 과정에서 각 장소 간의 승객 이동 소요시간을 각 게이트마다 고려하여 계산하였다.

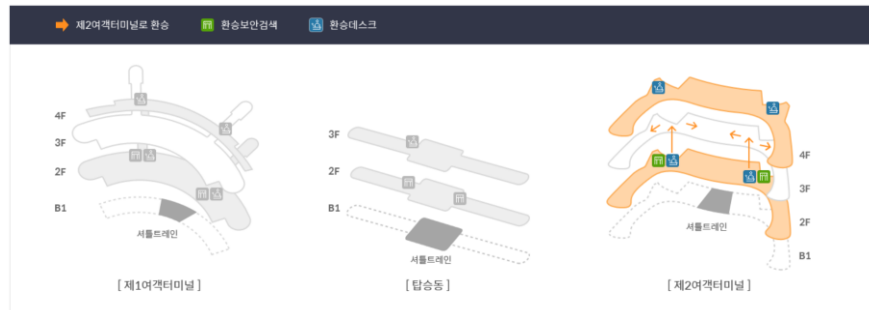
3) Terminal 2



- Terminal 2 -> Terminal 1: 도착 게이트 -> 셔틀트레인 -> 탑승동 도착 -> 셔틀트레인 -> 제1여객터미널 도착 -> 보안 검색장 -> 환승데스크(티켓) -> 출발 게이트



- Terminal 2 -> Concourse: 도착 게이트 -> 셔틀트레인 -> 탑승동 도착 -> 보안 검색장 -> 환승데스크(티켓) -> 출발 게이트



- Terminal 2 -> Terminal 2: 도착 게이트 -> 보안 검색장 -> 환승데스크(티켓) -> 출발 게이트
- 제2여객터미널에 도착한 승객들은 모두 탑승이 진행되는 터미널에서 보안 검색을 시행한다. 그러므로 어떤 터미널에서 탑승을 진행하는지에 상관없이 모든 루트는 '도착 게이트 -> 각 터미널로 이동 -> 보안 검색 -> 티켓 발급 -> 출발 게이트'의 순으로 진행된다. 따라서, 제2여객터미널에 도착한 모든 승객은 각자의 터미널로 이동한 후에 Security를 통과하고, 그곳에서 Ticketing Desk를 이용하도록 구성하였다. 티켓을 발급받은 승객들은 각자 출발을 위해 탑승할 게이트로 이동한다. 전반의 과정에서 각 장소 간의 승객 이동 소요시간을 각 게이트마다 고려하여 계산하였다.

3. 데이터 가정

1) 공항에 도착하는 승객

공항에 도착하는 승객의 데이터는 [인천국제공항공사 \(airport.kr\)](https://www.airport.kr)에서 제공하는 데이터¹⁶를 이용하였다. 터미널 당 승객 수도 확인할 수 있었는데, 제1여객터미널과 탑승동의 승객 수가 T1데이터로 한꺼번에 정리되어 있어 각 터미널의 게이트 수의 비율에 따라 전체 데이터에 5/8을 곱하여 제1여객터미널의 데이터로 활용하였고, 3/8을 곱하여 탑승동의 데이터로 활용하였다. 공항에 도착한 승객들의 수는 인천국제공항이 30일 동안 1시간 단위로 승객 이용 현황을 파악한 데이터를 기준으로 하루에 도착하는 시간 당 승객수를 구하였다. (아래 그림) 11월 한달 동안 특정 시간대에 공항에 도착한 승객의 수를 30으로 나누어 하루 중 특정 시간대에 몇 명의 승객이 도착하는지를 파악하였고, 이를 각 터미널의 게이트 수(T1: 47개, Con: 30개, T2: 40개)로 나누어 1시간 단위로 각 터미널의 한 게이트에서 생성되는 도착 승객수를 가정하였다. (이 과정으로 구한 수치는 csv파일(ICN_arrival.csv)로 첨부하였다.)

시간	Total Passenger	Passengers per Day	T1 Passenger	Con Passenger	T1 Passengers per Gate Round	Con Passengers per Gate Round	T2 Passenger	Passengers per Day	T2 Passengers per Gate Round
00:00 ~ 00:59	11,841	395	236.82	157.88	5.038723404	5	5,262,666,667	53	1,325
01:00 ~ 01:59	1,437	48	28.74	19.16	0.611489362	1	0.638666667	11	0.27
02:00 ~ 02:59	85	3	1.7	1.133333333	0.036170213	0	0.037777778	83	2.07
03:00 ~ 03:59	5,876	196	117.52	78.3466667	2.50042532	3	2.611555556	175	4.3675
04:00 ~ 04:59	49,405	1,650	989.9	659.9333333	21.06170213	21	21.99777778	1,080	27.00916667
05:00 ~ 05:59	91,368	3,046	1827.36	1215.24	38.88	39	40.608	41	75.549
06:00 ~ 06:59	87,979	2,933	1759.58	1173.053333	37.43787234	37	39.10177778	39	58.455
07:00 ~ 07:59	91,333	3,044	1826.66	1217.773333	38.86510638	39	40.59244444	41	75.549
08:00 ~ 08:59	58,987	1,966	1179.74	786.4933333	25.10085106	25	26.21644444	26	29.372
09:00 ~ 09:59	73,113	2,437	1462.26	974.04	31.11191409	31	32.49466667	32	19.173
10:00 ~ 10:59	51,219	1,707	1024.38	682.92	21.79531915	22	22.764	23	20.068
11:00 ~ 11:59	90,844	3,028	1816.88	1211.253333	38.65702128	39	40.37511111	40	31.241
12:00 ~ 12:59	68,359	2,279	1367.18	911.4533333	29.08893617	29	30.38177778	30	20.913
13:00 ~ 13:59	89,204	2,973	1784.08	1189.386667	37.95914894	38	39.64622222	40	46.017
14:00 ~ 14:59	94,330	3,144	1886.6	1257.733333	40.14042553	40	41.92444444	42	22.957
15:00 ~ 15:59	78,807	2,627	1576.14	1050.76	33.53489362	34	35.02533333	35	80.786
16:00 ~ 16:59	98,434	3,148	1888.68	1259.12	40.18460805	40	41.97066667	42	107.807
17:00 ~ 17:59	108,915	3,664	2198.3	1465.533333	46.77234043	47	48.85111111	49	117.878
18:00 ~ 18:59	103,318	3,444	2066.36	1377.573333	43.96510638	44	45.91911111	46	38.547
19:00 ~ 19:59	92,348	3,078	1846.96	1231.306667	39.29702128	39	41.04355556	41	41.920
20:00 ~ 20:59	100,298	3,343	2005.96	1337.306667	42.68	43	44.57688889	45	28.305
21:00 ~ 21:59	84,874	2,829	1697.48	1131.653333	36.11659574	36	37.72177778	38	23.786
22:00 ~ 22:59	52,644	1,755	1052.88	701.92	22.40170213	22	23.39733333	23	15.758
23:00 ~ 23:59	15,405	514	308.1	205.4	6.555319149	7	6.846666667	7	4.360
합계	1,597,513	53,250	31950.26	21300.17333	679.792766	680	710.0057778	710	860.911

앞서 언급했듯이 120여개에 달하는 인천공항 내의 모든 입국 게이트를 일일이 표현하기에는 무리가 있을 것이라고 판단하여 특정 게이트로부터 각 터미널에 위치한 보안 검색장까지의 거리를 기준으로 약 3-9개의 게이트를 묶어 통합 게이트를 구성하였다. 위의 과정을 통해 각 터미널의 한 게이트 당 도착하는 승객들의 시간별 데이터를 얻었으므로, 해당 데이터를 Schedule로 지정하여 각 터미널 당 한 시간에 도착하는 승객 수를 반영하였고, 통합 게이트 당 도착하는 승객의 시간별 분포를 나타내기 위해 Entities per Arrival을 통합 게이트를 구성하고 있는 게이트 수만큼 설정하여 한 게이트에서 발생하는 승객 수를 최대한 구현하였다.

¹⁶ 인천국제공항공사, <https://www.airport.kr/co/ko/cpr/statisticCategoryOfTime.do>

- 2) 각 게이트에 도착한 승객들이 다음 행선지로 이동하는 소요시간은 인천국제공항 공식홈페이지의 공항지도¹⁷를 참고하여 공항 내부 이동 시 소요시간을 바탕으로 한 삼각분포를 이용하였고, 이를 Walk Time 이라는 attribute로 지정하여 고정 매개변수로 사용하였다.
- 3) 전체 승객 중 환승객의 비율 또한 인천국제공항공사 (airport.kr)¹⁸의 데이터를 활용하였다.

인천공항 통계보기

시계열 통계	요일별통계	시간대별통계	지역별통계	항공사별통계	지선통계	결항통계	객화률통계	항공자료
--------	-------	--------	-------	--------	------	------	-------	------

운항: 인천, 항공사: 여객기, 노선: 국내선
 화물: 인천, 터미널: 국제, 여객: 환승승객
 날짜: 2023년 11월 ~ 2023년 11월

※ 인천공항 통계만 작성
 ※ 터미널별 실적은 여객기 운항 및 여객 실적만 수록 가능
 ※ 도착 총합은 인천 공항 기준
 ※ 해당 월 통계자료는 익월 10일(주말 공휴일 제외) 기준 이후 공표될 예정입니다.

인천공항 통계보기

시계열 통계	요일별통계	시간대별통계	지역별통계	항공사별통계	지선통계	결항통계	객화률통계	항공자료
--------	-------	--------	-------	--------	------	------	-------	------

운항: 인천, 항공사: 여객기, 노선: 국내선
 화물: 인천, 터미널: T1, 여객: 환승승객
 날짜: 2023년 11월 ~ 2023년 11월

※ 인천공항 통계만 작성
 ※ 터미널별 실적은 여객기 운항 및 여객 실적만 수록 가능
 ※ 도착 총합은 인천 공항 기준
 ※ 해당 월 통계자료는 익월 10일(주말 공휴일 제외) 기준 이후 공표될 예정입니다.

요일	인천			여객			화물		
	도착	출발	합계	도착	출발	합계	도착	출발	합계
월요일	1,753	1,739	3,492	26,507	36,312	72,819	4,291	5,033	9,293
화요일	1,695	1,639	3,334	26,193	37,449	73,642	4,315	5,288	9,604
수요일	2,166	2,275	4,441	47,216	47,776	94,992	6,469	7,900	14,369
목요일	2,171	2,117	4,288	45,543	43,998	89,541	6,148	6,900	13,048
금요일	1,773	1,779	3,552	25,195	34,459	59,654	4,706	6,047	10,753
토요일	1,733	1,736	3,469	24,740	34,238	58,978	4,431	5,838	10,269
일요일	1,813	1,836	3,649	32,337	33,334	65,671	5,000	5,929	10,929
합 계	13,104	13,121	26,225	269,731	267,566	537,297	33,359	42,904	76,264
전년대비 증감률	+85.5%	+95.3%	+90.4%	+46.5%	+45.2%	+45.8%	+41.8%	+57.8%	+50.1%

요일	인천			여객			화물		
	도착	출발	합계	도착	출발	합계	도착	출발	합계
월요일	1,197	1,193	2,390	12,258	12,525	24,783	1,193	1,193	2,386
화요일	1,130	1,121	2,251	11,296	11,363	22,659	1,195	1,581	2,776
수요일	1,495	1,581	3,076	14,546	15,291	29,837	1,496	1,499	2,995
목요일	1,215	1,214	2,429	10,631	10,901	21,532	1,190	1,206	2,396
금요일	1,257	1,257	2,514	10,908	11,838	22,746	1,257	1,257	2,514
토요일	1,190	1,206	2,396	11,847	12,189	24,036	1,190	1,206	2,396
일요일	1,257	1,257	2,514	10,908	11,838	22,746	1,257	1,257	2,514
합 계	9,002	9,011	18,013	85,362	87,766	173,128	9,002	9,011	18,013
전년대비 증감률	+81.7%	+81.6%	+81.6%	+92.3%	+93.0%	+92.6%	+81.7%	+81.6%	+81.6%

- ① 월별 인천공항에 도착하는 전체 승객의 수는 제1여객터미널이 160만명, 제2여객터미널이 86만명 정도이며, 환승을 위해 인천 공항에 도착하는 전체 터미널의 환승객 수는 2023년 11월 기준 일주일에 26만명 정도이다.
 - ② 그 중 제1여객터미널에 도착하는 환승객 수는 8만명, 그리고 제2여객터미널은 18만명 정도이다.
 - ③ 탑승동은 제1여객터미널에 포함되므로 게이트 개수(T1 50개, 탑승동 30개)를 고려하여 제1여객터미널에 도착하는 환승객의 수를 5만명, 탑승동의 환승객 수를 3만명이라고 가정했을 때,
 - ④ 전체 승객 대비 환승하는 승객의 도착 비율은 제1여객터미널과 탑승동이 8만/160만 X 100 = 5(%)이므로 각 터미널의 Transfer Decision의 비율을 5%로 설정하였다.
 - ⑤ 마찬가지로 제2여객터미널의 Transfer Decision 비율은 18만/86만 X 100 = 약 21(%)로 설정하였다.
- 4) **보안 검색 소요 시간**은 보안 검색대를 통과하는 데 걸리는 시간으로 이외에 짐을 놓거나 찾는 행위 역시 소요 시간에 포함하기로 하였고, 경험적인 수치를 바탕으로 초 단위의 TRIA(20, 35, 50) 분포를

¹⁷ 인천국제공항, <https://www.airport.kr/ap/ko/svc/airlinesTerInfoT1List.do>

¹⁸ 인천국제공항공사, <https://www.airport.kr/co/ko/cpr/statisticCategoryOfDay.do>

반영하였다.

- 5) **Decide Terminal의 비율**은 인천국제공항공사의 각 터미널의 출발 승객 수 데이터를 활용하였다.
 - ① 먼저, 인천공항 전체에서 출발한 환승객 수는 2023년 11월 기준 267,566명이며, 그 중 제1여객터미널에서 출발한 환승객은 87,766명이고, 제2여객터미널에서 출발한 환승객은 179,800명이다.
 - ② 비율로 계산하면 제1여객터미널에서는 $87,766/267,566 = \text{약 } 33(\%)$ 의 환승객이 출발하였고, 제2여객터미널에서는 $179,800/278,566 = \text{약 } 67(\%)$ 의 승객이 출발하였다.
 - ③ 탑승동은 제1여객터미널에 포함되므로 게이트 개수를 고려하여 제1여객터미널에 도착하는 환승객은 $33(\%) \times 5/8 = \text{약 } 21(\%)$, 탑승동에서 출발하는 환승객은 $33(\%) \times 3/8 = \text{약 } 12(\%)$ 로 설정하였다.
- 6) **셔틀트레인 Station까지 이동 및 탑승 대기**는 5분의 셔틀트레인 배차간격과 도보 거리를 고려하여 삼각분포를 가지는 Delay 모듈을 생성하였다. 우선 입국한 모든 환승객들은 Walk Time을 거쳐 보안 검색장으로 이동한 상황이므로 보안 검색장에서 셔틀트레인 정류장까지의 거리 및 소요시간만을 고려하였다. 이는 모든 터미널에서 공통적으로 비슷한 거리이고, 소요시간은 TRIA(5, 7, 10)분 정도이다. 여기에 셔틀트레인을 탑승하기까지의 대기시간을 modular 함수로 설정하고 두 시간을 더하여 총 소요시간을 $\text{TRIA}(5, 7, 10) + (5 - \text{MOD}(\text{TNOW}, 5))$ 의 분포로 설정하였다. 제1여객터미널과 제2여객터미널 간의 이동은 두번의 대기가 필요하므로 최대 배차간격을 10분으로 설정하여 $\text{TRIA}(5, 7, 10) + (10 - \text{MOD}(\text{TNOW}, 10))$ 로 설정하였다.
- 7) **셔틀트레인의 이동 시간**은 공항지도에 표기되어 있는 소요시간인 5분을 고려하여 TRIA(4, 5, 6)의 분포를 가지는 Route로 설정하였고, 제1여객터미널과 제2여객터미널 사이의 이동은 셔틀트레인을 두 번 이용하므로, TRIA(9, 10, 11)로 설정하여 두번의 이동을 한 번의 Route로 나타내었다.
- 8) 최종 탑승을 진행할 터미널로 모인 승객들이 **보안 검색장으로 이동**하는 데에 소요되는 시간과 보안 검색을 마치고 티켓을 발급받기 위해 **환승데스크로 이동**하는 시간은 각 터미널마다 보안 검색장에서 환승데스크로 이동하는 거리는 유사하기 때문에 이를 고려하여 동일하게 TRIA(5, 7, 10)의 삼각 분포를 이용하여 측정했다.
- 9) 또한, **환승데스크**에서 티켓을 발급받는 시간은 데스크의 직원의 인원

및 정보 확인 절차가 필요하므로 경험적인 수치를 바탕으로 TRIA(2, 2.5, 3)의 분포를 가정하였다.

10) **보안 검색장과 환승데스크의 Queue**에는 Resource를 적용하여 내부 규모의 Capacity를 표현해주었다.

① 보안 검색장에는 총 6개의 보안검색대가 있기 때문에 보안 검색대의 Resource에 6의 Capacity를 부여했다.

② 제1여객터미널의 환승데스크는 2곳이고 총 4개의 창구로 구성되어 있다. 그런데 포털의 여행 후기¹⁹에 따르면 각 게이트 앞 탑승구에 위치한 데스크에서도 발권을 받을 수 있다고 하여 환승데스크의 창구 수와 제1여객터미널의 게이트 수를 합친 55개를 두 파트로 나누었고, 이 중 80%정도가 이용될 것이라고 가정하여 Ticketing Desk의 Resource에 대한 Capacity를 $55/2 \times 0.8 = 22$ 로 설정하였다.

③ 탑승동의 환승 데스크는 1개이며 창구는 4개 존재한다. 또한 탑승동의 게이트 수는 30개이므로 총 34개의 데스크에서 발권이 가능하며, 이 중 80%정도가 이용될 것이라고 가정하여 Ticketing Desk의 Resource에 대한 Capacity를 $34 \times 0.8 = 27$ 로 설정하였다.

④ 제2여객터미널에 존재하는 환승데스크는 총 2개로 4개의 창구가 있다. 또한 제2여객터미널의 게이트수는 40개로 총 48개의 발권장소가 존재하므로 이를 두 파트로 나누었고, 이 중 80%정도가 이용될 것이라고 가정하여 Ticketing Desk의 Resource에 대한 Capacity를 $48/2 \times 0.8 = 19$ 로 설정하였다.

11) **최종 환승 게이트까지 이동하는 과정**에는 특정 승객이 최종 탑승하고자 하는 게이트를 별도로 지정하지 않고, 처음 입국한 모든 승객에게 할당했던 Walk Time을 이용하여 티켓 발급을 마치고 환승데스크에서 각자의 게이트로 향할 승객들의 도보 소요시간을 분석하였다. 각 터미널 별로 Walk Time의 최댓값과 최솟값을 UNIF함수로 설정하여 Route의 소요시간으로 부여하였다.

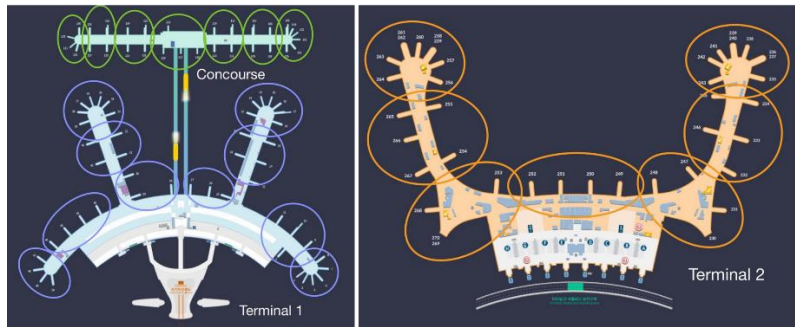
4. 모델 전체 세부 설명(흐름, 모듈단위까지)

1) Passenger Arrival: Create -> Assign(Attribute, Arrival Time = TNOW) -

¹⁹ 인천국제공항 국제선 환승 후기, <https://m.blog.naver.com/hannapride/221298041662>

> Decide

- ① Create(Gx to Gy, x, y은 게이트 번호): 각 터미널의 게이트로부터 보안 검색장까지의 거리를 고려하여 아래의 그림과 같이 그룹화하였고, 이를 하나의 통합 게이트로서 Ceate에 설정하였다. 그리고 하나의 Create 모듈에 속한 게이트의 수만큼 Entities per Arrival을 설정하여 해당 게이트에서 빠져나오는 승객의 수를 구현하였다.



- ② Assign(Assign Gx): 우선 Attribute로 Walk Time 설정하여 Create로 지정한 통합게이트로부터 보안 검색장까지의 도보 소요시간을 할당하였다.
- ✓ 제1여객터미널: 게이트와 보안 검색대와의 거리에 따라 가장 가까운 거리부터 TRIA(1, 3, 7), TRIA(1, 5, 7), TRIA(3, 5, 7), TRIA(7, 10, 13), TRIA(8, 10, 12), TRIA(13, 15, 20)까지 총 6가지의 Walk Time을 설정하여 10개의 게이트에 assign으로 할당하였다.
 - ✓ 탑승동: 게이트와 보안 검색대와의 거리에 따라 가장 가까운 거리부터 TRIA(1, 2, 3), TRIA(3, 4, 5), TRIA(5, 7, 9), TRIA(8, 10, 12)까지 총 4가지의 Walk Time을 설정하여 7개의 게이트에 assign으로 할당하였다. 탑승동은 보안검색을 거친 후 이동하는 승객들이 어느 보안 검색장을 이용할지에 대해 Position at Con이라는 Attribute를 정의하여, 그 값이 1(Right)이면 Security1을, 0(Left)이면 Security2를 이용하도록 하여 승객의 이동 동선을 재현하였다.
 - ✓ 제2여객터미널: 게이트와 보안 검색대와의 거리에 따라 가장 가까운 거리부터 TRIA(1, 2, 3), TRIA(1, 3, 5), TRIA(5, 7, 9), TRIA(8, 10, 12), TRIA(12, 13, 15)까지 총 5가지의 Walk Time을 설정하여 7개의 게이트에 assign으로 할당하였다.

또한 Arrival Time을 Attribute로 TNOW수치를 부여함으로 최종적으로 환승 시간을 기록하는 데에 이용하였다.

- ③ Decide(Transfer Decision for Gx to Gy): 앞서 구한 각 터미널마다의 환승 비율(T1: 5%, Con: 5%, T2: 21%)에 따라 전체 도착 승객의 일부가 환승하도록 Decide 모듈을 이용했다. 환승하지 않는 승객은 Dispose 모듈 (Non Transfer Passengers at Terminal)을 이용해 시뮬레이션 과정에서 빠져나가도록 하였다.

2) Terminal 1: Process(Security Check Queue) -> Decide -> Delay(Walk&Wait) -> Station & Route(to Transfer Desk)

제1여객터미널은 구조상 모든 게이트의 승객이 보안검색을 받은 후 셔틀트레인을 통해 각자의 터미널에 위치한 환승데스크로 이동하는 경로이므로 다음과 같은 모듈을 사용하였다.

- ① Process(Security): 제1여객터미널은 좌우 대칭인 구조로, 좌측에 위치한 게이트에서 모이는 승객들은 Security 1을, 우측에 위치한 게이트에서 모이는 승객들은 Security 2를 이용하기 때문에 2개의 Process모듈을 생성하여 각각 5개의 게이트씩 해당 Process로 향하도록 설계하였다. 이 Process를 통해 발생하는 Queue는 seize Delay and Release Type으로 Resource에 Capacity 6을 설정하여 검색대의 수를 표현하였으며, Delay time은 Expression을 생성하여 Security Time이 TRIA(20, 35, 50)을 따르도록 설정하였다.
- ② Decide: 제1여객터미널에서 보안검색을 마친 승객들이 어느 터미널로 이동할지를 결정해주는 모듈이다. 앞서 가정한 대로 각 터미널의 환승비율에 따라 N-way by chance Type을 설정하여 3가지의 갈래를 나눈 후 각각의 갈래에 12, 21을 부여한다. 그 후 12의 값을 가지는 갈래에 T1의 Process를, 21의 값을 가지는 갈래에 Con의 Process를, 그리고 else 갈래에 T2의 process를 연결하여 각 터미널로 이동하는 승객의 수를 구현한다.
- ③ Delay(Walk&Wait): 탑승동과 제2여객터미널로 이동하는 승객은 셔틀트레인에 탑승하기 위해 정류장으로 이동하여 배차시간에 맞추어 탑승 대기 시간을 갖는데 이때 소요되는 시간을 Delay모듈로 설정하였고, 이 시간은 도보 이동 시간 + 탑승 대기 시간의 값으로 TRIA(5, 7, 10) + (5 - MOD(TNOW,5))의 분포를 가정하였다.

- ④ Station & Route(to Transfer Desk): 실제 정류장을 표현하기 위해 Station모듈에 Name을 설정하여 나타냈고, 이 Station에서 출발하는 셔틀트레인의 경로를 Route로 표현하였다. 제1여객터미널로 향하는 경로는 셔틀트레인을 이용하지 않기 때문에 TRIA(0, 0, 0.5)로 설정하였고, 탑승동으로의 이동은 셔틀을 한 번 이용하기에 평균 셔틀 이동 시간이 5분인 데이터를 참고하여 TRIA(4, 5, 6)의 분포로 설정하였다. 제2여객터미널로의 이동은 셔틀을 2번 이용해야 하므로 이동시간이 10분이라고 가정하고, 분포를 TRIA(9, 10, 11)로 설정하였다. 그리고 제1여객터미널의 승객들은 이미 보안 검사를 마쳤으므로 각각의 Route는 해당 터미널의 Ticketing Desk에 해당하는 Station으로 이동하도록 하였다.

3) Concourse: Decide -> Process(Security Check Queue) -> Delay(Walk&Wait) -> Station & Route(to Security or Transfer Desk)

탑승동은 구조상 제1여객터미널로 이동하는 승객은 곧바로 셔틀트레인을 이용하여 이동한 후 그곳에서 보안검색과 티켓 발급을 진행한다. 반면에, 탑승동과 제2여객터미널에서 탑승을 진행하는 승객들은 탑승동에서 보안검색을 받은 후 각자의 터미널로 이동하여 그곳에서 환승데스크를 이용하는 경로이므로 다음과 같은 모듈을 사용하였다.

- ① Decide 1(Decide Terminal at Con 1): 탑승동에서 보안 검색을 받지 않는 제1여객터미널로 향하는 승객들과 나머지 승객들을 분리하기 위해 2-way by chance로 21% 비율의 갈래를 설정하여, 이를 제1여객터미널 승객의 경로와 연결하였다.
- ② Decide 2(Security Decision at ConCourse): 앞선 과정에서 탑승동의 각 게이트마다 좌, 우 위치에 관한 Attribute를 정의한 Assign을 할당해주었는데, Decide 2에서는 이 위치 정보에 따라 1의 값을 가지는 Right Position에 위치한 승객은 Security1로, 아닌 승객은 Security2로 보내는 2-way by condition을 활용하였다.
- ③ Process(Security 1, 2 at Con): 탑승동 역시 좌우 대칭인 구조로, 좌측에 위치한 게이트에서 모이는 승객들은 Security 1을, 우측에 위치한 게이트에서 모이는 승객들은 Security 2를 이용하기 때문에 2개의 Process모듈을 생성하였고, Position at Concourse의 Attribute가 1에 해당하는 게이트 3개는 Security1 Process로, 나머지 4개의 게이트는 Security2 Process향하도록 설계하였다. 이 Process를 통해 발생하는 Queue는 seize Delay and Release Type

으로 Resource에 Capacity 6을 설정하여 검색대의 수를 표현하였으며, Delay time은 Expression을 생성하여 Security Time이 $TRIA(20, 35, 50)$ 을 따르도록 설정하였다.

④ Decide 3(Decide Terminal at Con 2): 탑승동에서 보안검사를 마친 승객들을 탑승동에서 환승을 진행할 승객과 제2여객터미널로 이동할 승객으로 구분하기 위해 12%인 탑승동 환승객과 67%인 제2여객터미널 승객의 비율을 계산하였고, 최종적으로 15%의 2-way by chance 갈래를 설정해 두 그룹의 승객을 분리하였다.

⑤ Delay(Walk to Station 3, 4 and Wait): 제1여객터미널과 제2여객터미널로 이동하는 승객은 셔틀트레인에 탑승하기 위해 정류장으로 이동하여 배차시간에 맞추어 탑승 대기 시간을 갖는데, 이때 소요되는 시간을 Delay모듈로 설정하였고, 이 시간은 도보 이동 시간 + 탑승 대기 시간의 값으로 $TRIA(5, 7, 10) + (5 - MOD(TNOW,5))$ 의 분포를 가정하였다.

⑥ Station & Route(to Transfer Desk):

✓ To terminal 1: 아직 보안검사를 진행하지 않았기 때문에 제1여객터미널의 Security에 해당하는 Station으로의 이동을 Route로 설정하였고, 셔틀트레인을 타고 이동해야 하므로 소요시간은 $TRIA(4, 5, 6)$ 의 분포를 따르도록 하였다.

✓ else: 탑승동과 제2여객터미널로 향할 승객들은 이미 보안검사를 마쳤으므로 곧바로 해당 터미널의 Ticketing Desk에 해당하는 Station으로 이동하는 Route를 설정하였다. 이때, 탑승동은 이동을 진행하지 않으므로 $TRIA(0, 0, 0.5)$, 제2여객터미널로의 이동은 셔틀트레인을 이용하므로 $TRIA(4, 5, 6)$ 의 분포를 따르는 소요시간을 가정하였다.

4) Terminal 2: Decide -> Delay(Walk&Wait) -> Station & Route(to Security)

① Decide(Decide Terminal at T2): 제2여객터미널에서 보안검색을 마친 승객들이 어느 터미널로 이동할지를 결정해주는 모듈이다. 앞서 가정한 대로 각 터미널의 환승비율에 따라 N-way by chance Type을 설정하여 3가지의 갈래를 나눈 후 각각의 갈래에 12, 21을 부여한다. 그 후 12의 값을 가지는 갈래에 T1의 Process를, 21의 값을 가지는 갈래에 Con의 Process를, 그리고 else 갈래에

T2의 process를 연결하여 각 터미널로 이동하는 승객의 수를 구현한다.

- ② Delay(Walk to Station 5, 6 and Wait): 제1여객터미널과 탑승동으로 이동하는 승객은 셔틀트레인에 탑승하기 위해 정류장으로 이동하여 배차시간에 맞추어 탑승 대기 시간을 갖는데, 이때 소요되는 시간을 Delay모듈로 설정하였고, 이 시간은 도보 이동 시간 + 탑승 대기 시간의 값으로 $TRIA(5, 7, 10) + (5 - MOD(TNOW, 5))$ 의 분포를 가정하였다.
- ③ Station & Route(to Security): 실제 정류장을 표현하기 위해 Station모듈에 Name을 설정하여 나타냈고, 이 Station에서 출발하는 셔틀트레인의 경로를 Route로 표현하였다. 제2여객터미널로 향하는 경로는 셔틀트레인을 이용하지 않기 때문에 $TRIA(0, 0, 0.5)$ 로 설정하였고, 탑승동으로의 이동은 셔틀을 한 번 이용하기에 평균 셔틀 이동 시간이 5분인 데이터를 참고하여 $TRIA(4, 5, 6)$ 의 분포로 설정하였다. 제1여객터미널로의 이동은 셔틀을 2번 이용해야 하므로 이동시간이 10분이라고 가정하고, 분포를 $TRIA(9, 10, 11)$ 로 설정하였다. 그리고 제2여객터미널의 승객들은 보안검사를 진행하지 않았으므로 각각의 Route는 해당 터미널의 Security Point에 해당하는 Station으로 이동하도록 하였다.

5) Security Check: Station(To Security at T1, Con, T2) -> Delay(Walk Time) -> Process(Security Check Queue) -> Route(Walk Time to Gate)

- ① Station(Security at T1, Con, T2): 각 터미널의 Security에 해당하는 Station을 생성하여 도착 터미널에서 이동하여 보안검색을 진행하는 승객들을 Route를 통해 받아들일 수 있도록 설정하였다.
- ② Delay(Walk to Security at T1, Con, T2): 셔틀트레인에서 하차하여 각 터미널의 Security Point로 이동하기까지의 소요시간을 Delay모듈로 설정하였다. 제1과 제2여객터미널은 셔틀트레인 정류장과 보안 검색대 간의 거리가 유사하기 때문에 소요시간은 동일하게 $TRIA(8, 10, 12)$ 로 설정하였고, 탑승동은 다른 터미널에 비해 거리가 짧은 점을 고려하여 소요시간을 $TRIA(2, 3, 5)$ 의 분포를 따르도록 하였다.
- ③ Process(Security N, N은 Security의 번호): 모든 터미널은 2개의 보안 검색대가 좌측과 우측에 1개씩 위치해있기 때문에 셔틀트레인에서 내린 승객들은 무작위로 보안 검색대를 선택하게 된다.

이를 반영하기 위해 Decide모듈의 2-way by chance Type을 이용하여 2개의 Security가 각각 50%의 승객을 충당하도록 설계하였고, 각 터미널 별로 2개씩의 Process모듈을 생성하여 경로를 설정하였다. 이 Process를 통해 발생하는 Queue는 seize Delay and Release Type으로 Resource에 Capacity 6을 설정하여 검색대의 수를 표현하였으며, Delay time은 Expression을 생성하여 Security Time이 TRIA(20, 35, 50)을 따르도록 설정하였다.

- ④ Route(To Ticketing Desk at T1, Con, T2): 보안검사를 마치고 해당 터미널의 환승데스크로 이동하는 경로를 Route모듈을 이용해 구현하였다. 각 터미널의 보안 검색대부터 환승데스크까지의 도보 이동 시간과 Route의 목적 Station을 다음과 같이 설정하였다.

- ✓ Terminal 1: TRIA(5, 7, 10), Station - Ticketing at Terminal 1
- ✓ Concourse: TRIA(3, 5, 7), Station - Ticketing at Concourse
- ✓ TRIA(5, 7, 10), Station - Ticketing at Terminal 2

- 6) Transfer Desk: Station(Ticketing at Terminal 1) -> Delay(Walk Time to Ticketing Desk) -> Decide(Ticketing Desk Decision) -> Process(Ticketing Queue) -> Route(To Boarding Gate)

- ① Station(Ticketing at Terminal): 보안검사 후 티켓 발급을 위해 환승데스크로 모이는 승객을 받아들이기 위해 각 터미널의 Ticketing Desk를 Station으로 설정하였다.
- ② Decide(Ticketing Decision at T1, Con, T2): 탑승동의 환승데스크는 1개이지만, 제1여객터미널과 제2여객터미널은 두 곳의 환승데스크가 존재하기 때문에 승객들은 무작위로 데스크를 선택할 수 있다. 이를 반영하기 위해 Decide모듈의 2-way by chance Type을 이용하여 2개의 Ticketing Desk가 각각 50%의 승객을 충당하도록 설계하였다.
- ③ Process(Ticketing at Desk N, N은 Desk의 번호): 위 내용 중 3-10)의 가정을 활용하여 각 터미널의 Ticketing Desk 개수를 Capacity로 반영하였으며, 제1여객터미널은 22, 탑승동은 27, 제2여객터미널은 19로 설정하였다. 각 터미널 당 Process의 개수는 실제 환승데스크의 개수인 2, 1, 2로 가정하였고, 이 Process를 통해 발생하는 Queue는 seize Delay and Release Type으로 Resource에 Capacity를 설정하여 검색대의 수를 표현하였으며,

Delay time은 Expression을 생성하여 Ticketing Time이 TRIA(2, 2.5, 3)을 따르도록 설정하였다.

- ④ Route(To Boarding Gate at T1, Con, T2): 보안 검색 후 티켓 발권 까지 마쳤으므로 기내에 탑승하기 위해 탑승 게이트까지 걸어가 는 시간을 고려하였다. 승객의 입국 과정에서 생성한 Attribute 인 Walk Time의 최댓값과 최솟값을 참고하여, 각 터미널마다 각 기 다른 도보 이용 시간에 대해 UNIF함수를 따르는 분포를 설정하였다. 그 설정값과 Route의 목적 Station은 다음과 같다.

- ✓ Terminal 1: UNIF(1 , 20), Station - Walk to Gate at T1
- ✓ Concourse: UNIF(1 , 12), Station - Walk to Gate at Con
- ✓ Terminal 2: UNIF(1 , 15), Station - Walk to Gate at T2

7) Record MCT: Station(Walk to Gate) -> Record -> ReadWrite -> Dispose(Board)

- ① Station(Walk to Gate at T1, Con, T2): 입국에서부터 환승과정을 마치고 기내 탑승에 이르기까지의 시간을 측정하기 위해 각 터미널에서 탑승을 진행하는 모든 승객들을 불러들일 Station을 설정하였다.
- ② Record(Record Time): Record 모듈을 이용하여 환승 시간의 Arrival Time에 해당하는 Time Interval 수치를 계산하여, 환승시간에 대한 통계값을 구해보고자 하였다. TNOW와 Arrival Time을 비교하여 Time Interval이라는 Tally에 추가한다.
- ③ Readwrite(Write Output): Other Type의 Assignment를 설정하여 TNOW - Arrival Time에 해당하는 환승시간(Time Interval)을 측정하고, csv파일에 한 개의 Transferring Traveler Entity 당 Replication 번호(NREP)과 Transfer Time(Time Interval과 동일한 값)을 기록하여 이를 통계와 검증에 이용할 수 있도록 하였다.
- ④ Dispose(Boarding): 최종적으로 모든 Entity가 Dispose되면서 시뮬레이션이 종료된다.

5. 모델 검증 계획

- 총 30번의 replication 동안 도착하는 승객(Passenger)의 NumberIN의 평균이 실제 하루동안 도착하는 국제선 항공편 승객 수인 82567명(3.1)을 통해 도출)과 유사하면 모델이 검증된 것으로 판단할 것이다.

- 구체적으로, NumberIN이 [실제승객수*0.95, 실제승객수*1.05]의 구간 내에 있는 경우 모델이 실제와 유사하게 설계되었다고 판단할 것이다.

4. 수치적 실험

A. 실험 계획

- 각 모델에서 활용된 데이터와 시뮬레이션에서 이용된 방법은 앞 장에서 설명하였다.
- 공항 별로 MCT가 올바르게 설정되어 있는지 가설검정을 통해 검증해보았다. 각 Replication 별 환승시간의 95분위 수를 평균 내어, 이를 MCT와 비교하는 검정을 통해서 적절한 비율(95%이상)의 환승객들이 MCT 내로 환승이 가능한 것인지 확인해보았다.
- MCT의 적절성을 판단하기 위해서 여러 번 Replication을 실시하였고, 이를 통해 더 많은 불확실한 시나리오들을 고려할 수 있다고 판단하였다. 1번의 t-test에 따른 가설검정에 전체 개별 entity가 갖는 환승시간을 기록하여서 empirical한 분포를 통해 더욱 현실적인 상황에 대해서 MCT의 적절성을 판단해보고자 하였다.
- 각 공항 모델 별 시뮬레이션 시작 - 종료 기준 시각

	LAX	DFW	SIN	ICN
시작 시각	05:00	04:00	04:30	00:00
종료 시각	24:00	22:00	23:30	24:00

- 각 모델 별 리포트, csv output 파일 명은 위에서 3.A.에서 명시하였다.

B. 실험 결과

- 로스앤젤레스 국제공항(Los Angeles International Airport, LAX)

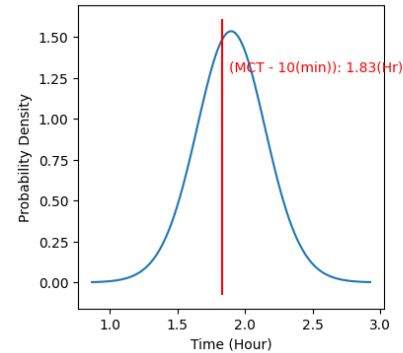
1. 모델 Verification

검증 결과, 30 replication 동안의 국제선 도착 승객의 Number in 평균은 26619.2로, 실제 데이터로 산정한 범위 [24177.5, 26722.5] 안에 속함을 확인할 수 있다. 따라서 해당 모델은 타당성을 갖추었다고 판단한다.

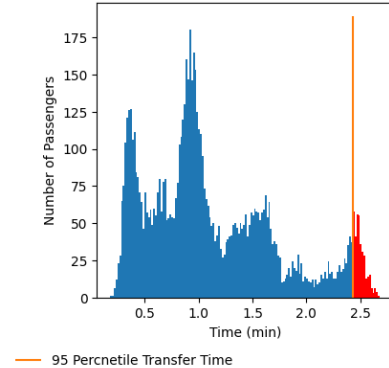
2. 결과 정리

REPLICATION	95% 환승시간 (hr)	REPLICATION	95% 환승시간 (hr)	REPLICATION	MCT보다 긴 환승시간 비율 (%)	REPLICATION	MCT보다 긴 환승시간 비율 (%)
1	1.9594	16	1.7154	1	0.1119	16	0.0286
2	1.9568	17	1.536	2	0.1246	17	0.0044
3	1.9833	18	2.0825	3	0.0719	18	0.1757
4	1.3788	19	1.789	4	0.0000	19	0.0356
5	1.5533	20	2.0171	5	0.0064	20	0.1852
6	2.5272	21	2.0327	6	0.2728	21	0.0898
7	1.8119	22	1.6961	7	0.0423	22	0.0178
8	1.7284	23	2.4296	8	0.0359	23	0.1373
9	1.8178	24	1.8211	9	0.0471	24	0.0453
10	1.8598	25	2.0548	10	0.0552	25	0.1423
11	1.8936	26	1.8775	11	0.0825	26	0.695
12	2.3987	27	1.8269	12	0.1838	27	0.0493
13	1.7334	28	1.6817	13	0.0133	28	0.0027
14	2.1262	29	2.0147	14	0.1251	29	0.1050
15	1.6276	30	2.008	15	0.0001	30	0.0911
평균		1.8979(시간)		평균		0.0784	

Los Angeles International Airport
t-Distribution for 95 Percentile Quantile Transfer Time



Los Angeles International Airport
Transfer Time Histogram of a Replication



LAX는 30번의 Replication에 대해 95%의 승객이 환승을 완료하는 데에 평균 1.8979시간이 소요되었으며, 이는 현재 LAX가 기준으로 삼고 있는 MCT 2시간에 비해서는 짧지만, 항공기 파이널 콜이 출발 10분 전에 이루어지는 것을 고려한 시간인 1.83시간보다는 조금 길다고 볼 수 있다. 30번의 Replication에 대해 MCT보다 긴 환승 시간이 소요된 승객의 비율은 평균적으로 0.0784(7.84%) 정도였으며, 이는 5%에 비해서는 다소 큰 값이다. 실제로 데이터를 확인한 결과 대부분의 승객은 0.3, 1.0, 1.5시간 근처에 환승을 완료하였지만 일부 승객이 1.9시간 또는 2.5시간 근처의 환승 시간을 소요한다는 것을 볼 수 있었다. T-test를 진행한 결과 (MCT - 10)과 95분위 수 승객의 환승 시간 평균값이 거의 일치한다는 결과를 얻었지만, MCT를 초과하여 환승하는 승객의 비율이 높다는 점은 여전히 주목해야 할 필요가 있다. (우측 그림)

ii. 댈러스-포트워스 국제공항(Dallas-FortWorth International Airport, DFW)

1. 모델 Verification

본 모델의 시뮬레이션을 통해 얻은 DFW 국제공항의 출발/탑승 승객의 인원 수를 실제 데이터와 비교하여 검증하였다. 마지막으로, 실제 데이터를 이용해 값을 추정함으로써 시뮬레이션 모델이 실제 상황과 최대한 유사하도록 구현해냈다. 특히 실제 데이터가 존재하는 TSA의 시뮬레이션 결과

얻을 수 있는 Wait Time을 주기를 실제 실시간 데이터와 비교함으로써 검증 하였으며, 타당하다는 평가를 내렸다. 추가적으로 ENTITY가 생성되는 숫자를 실제 데이터의 수치와 비교해봄으로써 시뮬레이션 과정이 실제 공항의 승객들로 발생하는 혼잡도를 추정할 수 있다고 보았다. (DFW 국제공항 설명의 3.8)

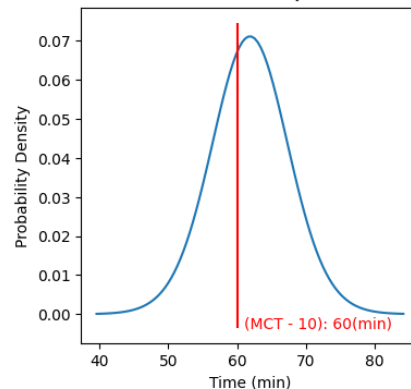
실제 터미널 D에 도착하는 탑승객의 숫자는 일평균 15186이었으며, Half-Width를 바탕으로 한 신뢰구간은 [14868, 15842] 이었다. 따라서, 실제 발생 수를 잘 반영하여 적절하게 시뮬레이션이 수행되었다고 판단하였다.

2. 결과 정리

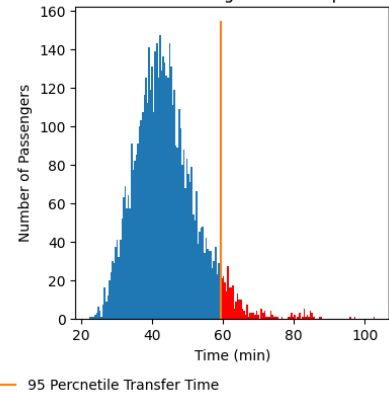
REPLICATION	환승시간의 95분위수 (min)	REPLICATION	환승시간의 95분위수 (min)
1	61.10	16	61.98
2	62.40	17	58.98
3	60.68	18	59.32
4	85.05	19	58.08
5	57.47	20	55.98
6	64.55	21	56.72
7	69.22	22	66.38
8	64.10	23	59.50
9	59.42	24	60.70
10	61.29	25	56.87
11	57.85	26	61.30
12	63.84	27	56.93
13	70.16	28	63.11
14	61.05	29	61.66
15	60.60	30	59.06
평균		61.86	

REPLICATION	MCT보다 긴 환승시간의 비율(%)	REPLICATION	MCT보다 긴 환승시간의 비율(%)
1	6.08	16	6.47
2	7.19	17	4.26
3	5.56	18	4.33
4	31.26	19	4.09
5	3.03	20	1.67
6	8.88	21	2.52
7	13.38	22	10.82
8	9.09	23	4.57
9	4.58	24	5.63
10	6.08	25	2.70
11	3.43	26	6.09
12	8.09	27	2.62
13	15.42	28	8.25
14	6.02	29	6.68
15	5.48	30	4.29
평균		7.00	

Dallas/Fort Worth International Airport
t-Distribution for 95 Percentile Quantile Transfer Time



Dallas/Fort Worth International Airport
Transfer Time Histogram of a Replication



DFW는 30번의 Replication에 대해 95%의 승객이 환승을 완료하는 데에 평균 61.86분이 소요되었으며, 이는 현재 DFW가 기준으로 삼고 있는 MCT 70분에 비해서는 짧지만 항공기 파이널 콜이 출발 10분 전에 이루어지는 것을 고려하면 60분보다는 조금 길다고 볼 수 있다. 30번의 Replication에 대해 MCT보다 긴 환승 시간이 소요된 승객의 비율은 평균적으로 7% 정도였으며, 이는 5%에 비해서는 다소 큰 값이다. 실제로 데이터를 확인한 결과 대부분의 승객은 40분 근처에 환승을 완료하였지만 일부 오른쪽 꼬리에 해당하는 승객이 1시간 이상의 환승 시간을 소요한다는 것을 볼 수 있었다. T-test를 진행한 결과 (MCT - 10)과 95분위 수 승객의 환승 시간 평균값이 거의 일치한다는 결과를 얻었지만, MCT를 초과하여 환승하는 승객의 비율이 높다는 점은 여전히 주목해야 할

필요가 있다. (오른쪽 그림)

iii. 창이 국제공항 (Singapore Changi International Airport, SIN)

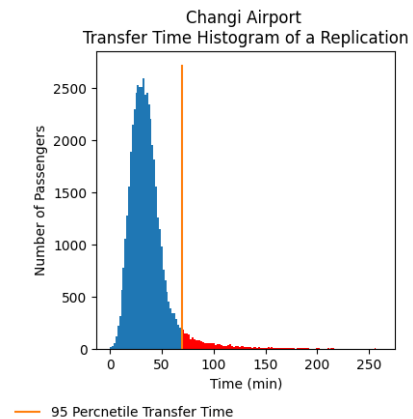
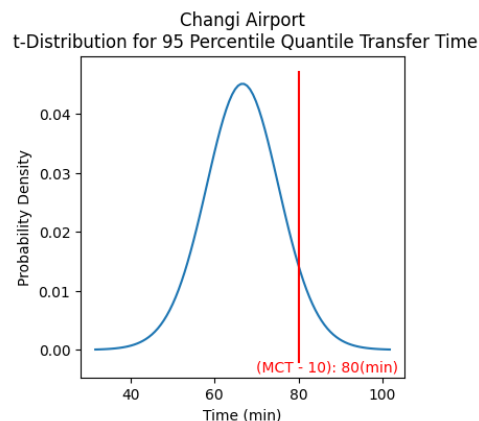
1. 모델 Verification

검증 결과, 30 replication 동안의 국제선 도착 항공기의 평균이 실제 데이터로 산정한 범위 안에 속함을 확인할 수 있다. 따라서 해당 모델은 타당성을 갖추었다고 판단한다.

2. 결과 정리

REPLICATION	95% 환승시간	REPLICATION	95% 환승시간
1	61.2174	16	64.6512
2	64.7884	17	55.0062
3	66.2629	18	77.3455
4	56.5840	19	60.3596
5	65.9248	20	89.4629
6	73.0275	21	73.8731
7	68.9173	22	69.5275
8	54.0619	23	61.3219
9	70.4230	24	61.8359
10	53.5542	25	66.8636
11	84.4165	26	67.3970
12	54.6660	27	56.0910
13	71.8982	28	65.7611
14	70.2071	29	70.1237
15	78.7813	30	67.0549
평균		66.7135(분)	

REPLICATION	MCT보다 긴 환승시간 비율 (%)	REPLICATION	MCT보다 긴 환승시간 비율 (%)
1	0.0135	16	0.0485
2	0.0257	17	0.0260
3	0.0321	18	0.0049
4	0.0094	19	0.0470
5	0.0307	20	0.0154
6	0.0309	21	0.0606
7	0.0408	22	0.0427
8	0.0338	23	0.0337
9	0.0037	24	0.0152
10	0.0352	25	0.0180
11	0.0003	26	0.0298
12	0.0545	27	0.0298
13	0.0013	28	0.0043
14	0.0370	29	0.0282
15	0.0358	30	0.0333
평균		0.0274	



SIN는 30번의 Replication에 대해 95%의 승객이 환승을 완료하는 데에 평균 66.7135분이 소요되었으며, 이는 현재 SIN이 기준으로 삼고 있는 MCT 90분에 비해서는 크게 짧고 항공기 파이널 콜이 출발 10분 전에 이루어지는 것을 고려하더라도 충분한 여유 시간이 있다고 볼 수 있다. 30번의 Replication에 대해 MCT보다 긴 환승 시간이 소요된 승객의 비율은 평균적으로 2.74% 정도였으며, 이는 5%에 비해 큰 여유가 있는 값이다. 실제로 데이터를 확인한 결과 대부분의 승객은 30분 근처에 환승을 완료하였고, 극소수 오른쪽 꼬리에 해당하는 승객이 80분 이상의 환승 시간을 소요한다는 것을 볼 수 있었다. T-test를 진행한 결과 (MCT - 10)의 값보다 95분위 수 승객의 환승 시간 평균값이 유의미하게 작다는 결과를 얻었고, 앞선 LAX와 DFW와는 달리 MCT가 넉넉하게 설정되어 있음을 확인할 수 있었다. 이는 싱가포르의 특성상 환승객이 큰 비율을 차지하

기 때문에 환승 실패로 인한 손해를 막기 위해서 항공권 판매량에서의 약간의 불이익을 감안하더라도 확실히 보장되는 환승을 제공하기 위해 노력하고 있는 것으로 볼 수 있다. (아래 그림)

iv. 인천 국제공항(Incheon International Airport, ICN)

1. 모델 Verification

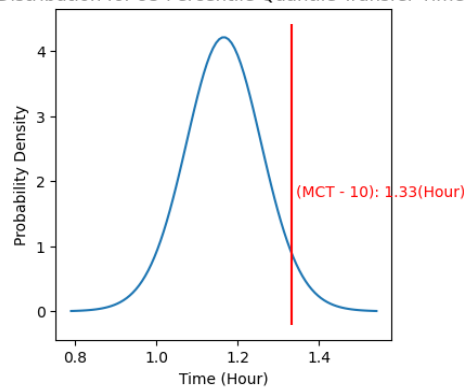
검증 결과, 30 replication 동안의 국제선 도착 승객(Passenger)의 NumberIN 평균은 81985.33으로, 실제 데이터로 산정한 범위 [78438.87, 86695.59] 안에 속함을 확인할 수 있다. 따라서 해당 모델은 타당성을 갖추었다고 판단한다.

2. 결과 정리

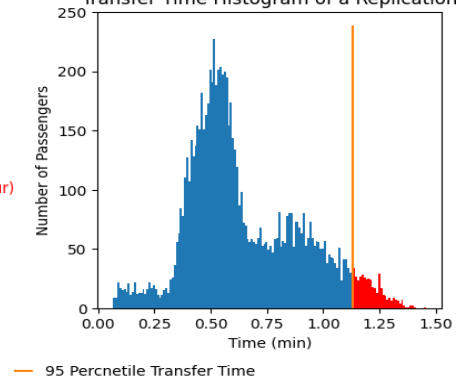
REPLICATION	95% 환승시간	REPLICATION	95% 환승시간
1	1.2127	16	1.2015
2	1.3692	17	1.1258
3	1.1081	18	1.0193
4	1.3308	19	1.1728
5	1.1337	20	1.0270
6	1.0704	21	1.2191
7	1.2138	22	1.1163
8	1.2223	23	1.1309
9	1.2071	24	1.1446
10	1.1509	25	1.0160
11	1.1981	26	1.2933
12	1.1075	27	1.2442
13	1.1507	28	1.2414
14	1.3346	29	1.0819
15	1.0907	30	1.0365
평균		1.1657(시간)	

REPLICATION	MCT보다 긴 환승시간 비율	REPLICATION	MCT보다 긴 환승시간 비율
1	1.5900	16	1.5500
2	5.9599	17	0.8399
3	0.2899	18	0.0000
4	4.9299	19	2.0299
5	0.9399	20	0.0000
6	0.0500	21	2.3700
7	1.6399	22	0.3599
8	2.4300	23	0.2700
9	2.2999	24	0.6600
10	0.6899	25	1.0000
11	1.7099	26	3.7900
12	0.6300	27	3.2099
13	0.6899	28	2.4300
14	5.0199	29	0.1300
15	0.2099	30	0.0000
평균		1.5906	

Incheon International Airport
t-Distribution for 95 Percentile Quantile Transfer Time



Incheon International Airport
Transfer Time Histogram of a Replication



ICN은 30번의 Replication에 대해 95%의 승객이 환승을 완료하는 데에 평균 1.1657시간(69.942분)이 소요되었으며, 이는 현재 ICN이 기준으로 삼고 있는 MCT 90분에 비해서는 크게 짧고 항공기 파이널 콜이 출발 10분 전에 이루어지는 것을 고려하더라도 충분한 여유 시간이 있다고 볼 수 있다. 30번의 Replication에 대해 MCT보다 긴 환승 시간이 소요된 승객의 비율은 평균적으로 1.59% 정도였으며, 이는 5%에 비해 큰 여유가 있는 값이다. 실제로 데이터를 확인한 결과 대부분의 승객은 30분(0.5시간) 근처에 환승을 완료하였지만,

일부 오른쪽 꼬리에 해당하는 승객이 80분 이상의 환승 시간을 소요한다는 것을 볼 수 있었다. T-test를 진행한 결과 (MCT - 10)의 값보다 95분위 수 승객의 환승 시간 평균값이 유의미하게 작다는 결과를 얻었고, 앞선 LAX와 DFW와는 달리 MCT가 넉넉하게 설정되어 있음을 확인할 수 있었다. 이는 인천공항의 특성상 국제선 환승객의 비율이 상대적으로 더 크고 환승객의 대다수가 외국인이라는 점을 고려하여, 이들 환승객들이 환승 프로세스를 순조롭게 진행할 수 있는 충분한 시간을 MCT로 선정한 것이라 해석해볼 수 있다. 또한, 승객 편의성, 운영 효율성, 경쟁력 강화 등 다양한 편익을 위해 실정에 맞는 최소 환승 시간을 제공할 수 있도록 노력하는 것이라고 볼 수 있다. (위의 그림 참조)

5. 결론

A. 결과 요약 및 분석

i. MCT의 적정 여부 판단 기준 - 가설 검정을 바탕으로

각 공항 모델별로 수행한 30개의 Replication 별로 얻을 수 있는 환승시간의 95 분 위수를 이용하여 가설 검정을 수행하였으며 수립한 가설은 아래와 같다:

귀무가설: 환승시간의 95분위 수 평균값이 MCT보다 작다.

대립가설: 환승시간의 95분위 수 평균값이 MCT보다 크다.

- 귀무가설을 기각할 수 없는 경우에는 환승객들의 95%이상이 MCT내로 환승을 가능하다는 것이 보장되므로, 설정된 MCT 값이 일반적인 상황에서 환승을 보장하는 시간이라 볼 수 있다.
- 귀무가설을 기각할 수 있는 경우에는 환승객들의 95%가 MCT내로 환승을 할 수 없기 때문에 일반적으로 MCT를 기준으로 해서 환승을 보장할 수 없다.

이러한 가설을 검증함으로써 MCT의 적절성을 판단하였다.

이에 더해, 만약 귀무가설을 기각할 수 있는 경우 시뮬레이션을 통해 구해낸 95분 위 수의 평균값을 새로운 MCT로 설정할 수 있을 것이라 판단할 수 있다.

ii. 구체적인 공항 모델별 가설 검정

공항	LAX	DFW	SIN	ICN
P value	0.09	0.04	약 1.00	약 1.00

세 개 공항(LAX, SIN, ICN)에서는 귀무가설을 기각할 수 없었으며, 이들 공항이 정한 MCT는 환승객들의 환승 항공편을 예약함에 있어서 적절한 가이드라인으로써 이용될 수 있음을 검증할 수 있었다. 이는 공항이나 항공사가 환승 스케줄을 효율적으로 운영하고 승객들에게 안정적인 환승을 제공하고 있다는 긍정적인 결과로 해석된다.

그러나, DFW 국제공항의 경우에는 귀무가설을 기각할 수 있어, 5%가 넘는 승객

들이 환승 과정에서 MCT 보다 긴 시간을 소요한다고 볼 수 있었다. 이는 해당 공항의 환승 프로세스나 MCT 설정에 개선이 필요할 수 있음을 시사하며, 이는 항공사와 공항 운영자에게 환승 프로세스나 MCT 설정에 대한 주의 및 개선을 요구하는 결과로서 작용한다.

SIN와 ICN은 pvalue가 상당히 높은 수치로 나타났다. 이들 공항은 많은 국제선 환승객들이 이용한다는 점을 바탕으로 하였을 때, 이들이 환승 프로세스를 순조롭게 이해하고 진행할 수 있도록 충분한 시간을 확보한 것이라고 해석해볼 수 있다. 추가적으로, 귀무가설을 기각할 수 있었던 DFW 경우에는 이들 승객이 원활한 환승을 경험할 수 있도록 MCT를 적절하게 반영 및 수정할 필요가 있다고 판단할 수 있었다. 즉, 환승객들에게 부과되는 불편을 최소화하기 위해 통계적으로 MCT와 환승 성공률 간에 유의미한 차이가 발생하지 않도록 환승 정책이나 MCT 설정에 대한 지속적인 업데이트가 필요하다.

B. 의의 및 한계

i. 의의

- 본 프로젝트는 기존에 항공 산업 곳곳에서 활용되고 있으나 명확히는 알려지지 않았던 공항에서의 최소환승시간(MCT)의 유의성을 검증하고자 시도하였고, 통계적 유의성 검증을 통해 해당 MCT의 가치를 재인식할 수 있는 기회를 창출하였다는 점에서 의의가 있다.
- 또한, 그 과정에서 공항의 환승 절차를 queuing system의 관점에서 인식하고 이에 불확정성을 부여하는 시뮬레이션을 활용하였다는 점에서도 의의가 있다.
- 아울러 하나의 공항의 사례만을 통해 MCT를 검증한 것이 아니라, 아시아권 공항 2곳과 미국 공항 2곳을 선정하여 시뮬레이션 모델링함으로써 서로 다른 공항의 다양한 특성을 인식하고 이들 각각에서의 MCT를 살폈다는 부분에서의 가치도 지니고 있다.

ii. 한계

- 그럼에도 현실의 상황을 온전하게 정확히 반영해내지는 못하였다는 한계를 지닌다. 실제 공항의 구조, 도착 항공편 및 환승객 수, 보안 검색과 입국 심사 등에서 소요되는 시간, 보안검색대 및 입국심사 카운터의 개수 등에 관한 정확한 정보를 확보하는 것에 무리가 있었다. 이러한 정보를 정확하게 반영하는 것이 아니라 가정하고 모델링을 진행하였기에 모델이 현실과 온전히 유사하다고 보기는 어려운 것이 사실이다. 특히, 4.B.에 있는 그림을 보면 LAX의 경우 승객의 도착에 관한 실제 데이터를 그대로 정확히 반영하지 못하였음을 짐작할 수 있다.
- 또한, 실제 공항의 현장에서는 항공사별로 환승에 소요되는 시간에 차이가 존재하기도 한다. 그러나 본 프로젝트에서는 항공사에 따른 차이는 고려하지

않아 이를 반영하지 못하였다는 한계를 가진다.

- 더불어, 실제 국제공항에서는 여러 유형의 환승이 있다. 국제선-국제선, 국제선-국내선, 국내선-국제선, 국내선-국내선의 총 4가지 유형의 환승이 있지만, 본 프로젝트에서는 그 중 일부만을 고려하였다. 따라서 여러 환승 유형이 겹침으로 인해 생기는 유동 인구 및 소요 시간 증가를 반영하지 못한 측면이 있다.

C. 현실에의 적용

- 현실에의 적용에서, 본 프로젝트는 MCT의 명확한 측정 기준을 정립하는 것에 도움을 줄 수 있을 것이다.

- 현재 MCT를 측정하는 명확한 기준(어떠한 절차를 포함시킬 것인지, 절차 당 소요 시간은 어떻게 설정할 것인지 등)이 미비한 상황이다.

- 이러한 배경에서, 본 프로젝트는 현실에 적용됨으로써 실제 상황과 프로젝트의 모델이 얼마나 유사한지를 확인하고, 유사한 부분이 있으면 해당 요소를 실제 MCT 측정 기준에 반영할 수 있을 것이다. 혹은 모델 중 현실과 유사하지 않은 부분이 있다면, 실제 MCT 측정 기준에서는 해당 요소를 반영하지 않도록 지침을 제공할 수 있다.

- 또한, 본 프로젝트 결과는 항공 관련 기관이나 공항 운영자에게 MCT를 개선을 제안하고자 할 때 유용한 정보로 활용될 수 있다. 프로젝트에서 도출된 시뮬레이션 결과와 현실의 차이를 분석함으로써 개선이 필요한 부분을 식별하고 조치를 취하도록 유도할 수 있다.

- 현실에 적용 시, 본 프로젝트의 결과는 항공편 예약 및 환승 스케줄링에 참고 자료로 활용될 수 있다. 이를 통해 승객들은 보다 정확하고 효율적인 환승 시간을 예측하여 여행 계획을 세울 수 있다. 또한, 항공 업계에서는 시뮬레이션을 통해 도출된 정보를 기반으로 서비스 품질을 향상시키는데 활용할 수 있다.

D. 프로젝트 심화 제안

- 본 프로젝트의 한계점을 보완하여 보다 발전된 연구를 진행하는 것은 가치 있을 것이다. 특히 이에서는 보다 엄밀한 데이터 수집 및 활용을 통해 실제와 흡사한 시뮬레이션 모델을 구현해 내기를 기대한다. 본 프로젝트와 같이 하나의 데이터를 변형하여 다방면적으로 활용하는 방안을 넘어, 크롤링 등을 통해 실제 데이터를 직접 수집함으로써 연구를 발전시킬 수 있을 것이다.

- 또한, 시뮬레이션 모델 내 혹은 외부적인 분석 기법을 활용하여 MCT를 초과하는 환승시간이 발생한 경우, 어느 구간에서 이러한 예외적인 소요가 발생했는지 탐구하는 것도 유익한 연구가 될 것이다. 이를 통해 특정 지점에서의 환승 프로세스나 인프라 문제를 파악하고 개선점을 찾을 수 있다.

- 추가적으로, 실제 데이터를 활용하는 경우 데이터의 품질과 신뢰성을 보장하기 위한 데이터 정제 및 검증 과정을 강화하는 것이 필요하다. 이는 연구 결과의 정확성을 높이는 데 도움이 될 것이다.