FLIGHT DATA MANAGEMENT AND SCHEDULING

Manoj Middepogu (160050075), Ritesh Goenka (160050047)
Sathvik Kollu Reddy (160050077), Saiteja Nangunoori (160050089)

August 7, 2019

Aviation is a fast growing industry. Consequently, we require fast and scalable solutions for flight database management and scheduling airside operations. The problem of scheduling various airside operations and resource allocation for the same is complex due to highly dynamic nature of the system, and thus we study only some specific components of this system. The primary objective of this project is to build a flight database management system, and use dynamic scheduling algorithms for resource allocation. We consider the problem of allocation of three types of resources, namely gates, runways and human resources.

1 Functional Specifications

Provided the scheduled times of flights and resources available at the airport (such as gates, runways and human resources), we try to efficiently allocate resources and schedule the flights at the airports. The objective is to minimize the cumulative passenger unhappiness function which in the next section. The number of human resources at the airport continuously varies. Also, the arriving flights may be delayed (due to bad weather conditions etc.) or they may be diverted (due to low fuel levels etc.). We deal with the above mentioned dynamic changes in the system by adjusting the allocated resources.

2 Resource Allocation Problem Statement

We attempted to create a general optimization problem, in which we obtained a term which is inversely proportional to the number of resources allocated to the flight. Even if one considers discrete time instants (say minute boundaries), the search space of the optimization problem is too large. In this case, one may use genetic or simulated annealing algorithms; but the run time of these algorithms are too large for practical use since we need to run the algorithm after constant short time intervals. Thus, we decide to break the problem into the following stages:

1. Gate Allocation: This refers to allocation of airport gates to arriving and departing flights.
2. Loading and Unloading: For departure, this refers to loading luggage (unloading for arrivals) and boarding passengers (alighting for arrivals) onto the aircraft. Human resources are allocated to flights during this stage.
3. Runway Allocation: This refers to allocation of airport runways to arriving and departing flights.

To deal with the above issues, we optimize each of these three stages separately instead of optimizing the system. The optimization objective is to minimize the cumulative passenger unhappiness function, $f = \sum_{F \in flights} l(F) \cdot n(F)$, where $l(F)$ is the delay in the passenger exit time due to the stage in consideration and $n(F)$ is the number of passengers in the flight.

3 Algorithm

We use three different algorithms for the three stages mentioned in Section 2. We use a queue to push a flight from one stage to another. The position of a flight in the queue is stored in the flight table. Time has been divided into slots, which are discrete time intervals in which an event happens. The slot time for the three algorithms is equal to 60 minutes, 5 minutes and 2 minutes respectively. The three algorithms are run periodically after an interval of 1 minute, 5 minutes and 2 minutes respectively. The sequence of stages is gate allocation followed by loading followed by runway allocation for departure flights; and runway allocation followed by gate allocation followed by unloading for arrival flights.

3.1 Gate Allocation

In this stage, we allocate gates to all flights that have 'NOGATE'(departure flights) or 'LANDED' (arrival flights) status. 'NOGATE' status of a flight indicates that it is a departure flight waiting for gate and 'LANDED' indicates that it is a arrival flight which is landed but is waiting for a queue. A gate will be allocated to a departure flight only after 60 minutes before its scheduled gate departure time. If there are fewer no of gates than no of flights, gates are allocated to flight with more number of passengers. The mathematical idea behind this prioritization is given below.
We can simplify our problem as to decide which flight has to be given preference if two flights are racing for the same gate. The lateness cost if f_1 is given gate before f_2 is $(60 - t_1) * p_1 + (120 - t_2) * p_2$ and $(60 - t_2) * p_2 + (120 - t_1) * p_1$ if f_2 is given gate before f_1 where t_i and p_i represents departure time and number of passengers of flight f_i respectively. We can say that from above equations f_1 should be given gate before f_2 if $p_1 > p_2$ and vice-versa.
The gate allocation algorithm performs the following sequence of steps:

- Collects all the free gates(gates not assigned to any other flight) in the airport and stores them in a list.
- Iterates over all flights which have 'NOGATE' or 'LANDED' status and pushes them in the gate allocation queue if their scheduled departure time is within 60 minutes from current time.
- Prioritizes the flights in above queue based on the no of passengers in them .
- Allocates all free gates to flights with high priorities by changing their status to 'GATE'.

The flights with status 'GATE' are now dealt in second stage in they are assigned resources to load/unload baggage and to board/get-off passengers. The resources here reflect to working crew and vehicles in real life.

3.2 Loading and Unloading

The number of resource time units required by a flight is equal to the number of resources to be allocated to finish the work in one slot time period. If R is the number of resource time units required by a flight, x is number of resources allocated to it and T is the slot time, then the time required to complete loading/unloading is equal to $R/x \cdot T$ and it's cost is $((R/x) \cdot T - t) * p$, where t is the departure time and p is the number of passengers in the flight. Let us consider the case in which there are only two flights. If k is the number of resources available at start of slot and we allocate x resources to the first flight, then the lateness of the two flights will be equal to $((R_1/x) \cdot T - t_1) * p_1$ and $(R_2/(k-x) \cdot T - t_2) * p_2$ respectively. On minimizing their sum w.r.t x, we obtain $x = k \cdot \sqrt{p_1 \cdot R_1}/(\sqrt{p_1 \cdot R_1} + \sqrt{p_2 \cdot R_2})$. Thus, we obtain the result that the number of resources that should be allocated to a flight is proportional to $\sqrt{p \cdot R}$. So at each run we allocate resources based on number of resource time units required by the flights with status 'GATE'. When no more resource time units are required, we push the flight into third stage and assign 'RUNWAY' status in case of departure.

3.3 Runway Allocation

A runway is assigned to a flight based on the number of flights already waiting at the various runways. We maintain a queue of flights waiting for departure at each runway on the airport. The runway with least queue length at the given instant is assigned to the flight. Flights waiting in a queue at a given runway are scheduled on the basis of FIFO principle. During each runway slot time, the flight at the front of each runway queue is selected and is scheduled on that corresponding runway. Arrival flights are handled in a similar manner as above by maintaining an extra queue at each runway(called arrival queue). The collision between arrival and departure flights is prevented by giving higher priority to the former as compared to the latter. After completion of the runway stage, the status of flight is changed to 'TAKEOFF' and 'LANDED' in case of departure and arrival respectively.

4 Users and Interfaces

We provide a web application interface with the following three types of users:

1. Public Users (passengers etc.)
    - The user will be able to view the status of all scheduled flights, on a specified date and route, on the homepage of the website.
    - No Special privileges are given to the user and no login is required.
2. Airport Authorities
    - The user has the privilege to change the number of human resources available at the airport and flight details such as destination (in case of a diversion) and estimated runway arrival time (in case of a delay).
    - The user can view details of flights scheduled at the airport in the next six hours, number of human resources, runways and gates.

- Login is required for these users.

3. Airlines
   - The user has the privilege of adding a scheduled flight to the database.
   - Login is required for these users.

5 Data and Testing

The following are the two types of inputs to our simulation:

- Details of scheduled flights given by various airlines.
- Resource and flight updates given by the airport authorities.

We use the flight data available at the website of United States Federal Aviation Administration. We choose United States domestic flight data for a single day (i.e. 11 Jan 2008). We change the date in the data to current date for generating a real-time simulation. We simulate the dynamic flight arrival and departure patterns using the scheduling framework as described earlier.

6 Entity Relationship Diagram and Relational Schema

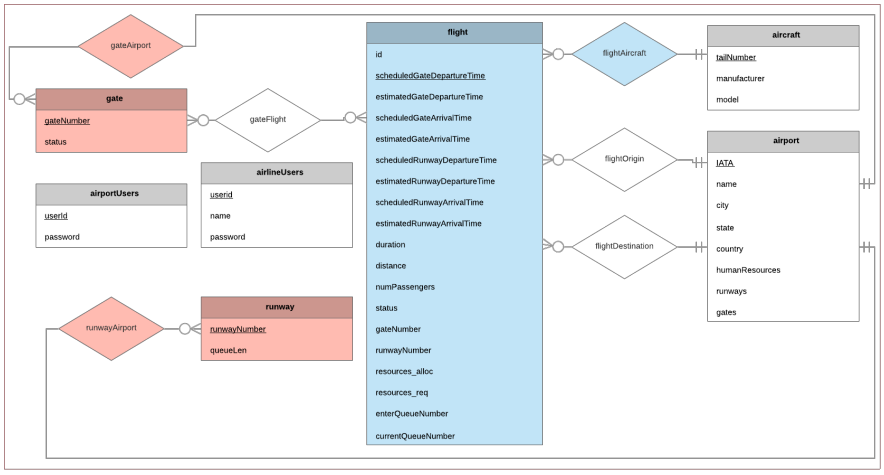

Figure 1: **Entity Relationship Diagram** Weak entities and their respective identifying relationships have been shaded with the same color. Discriminator attributes of these weak entities have been underlined. Keeping in mind the efficiency of the dynamic scheduling algorithm to be run at the local airport server, some redundant attributes have been added.

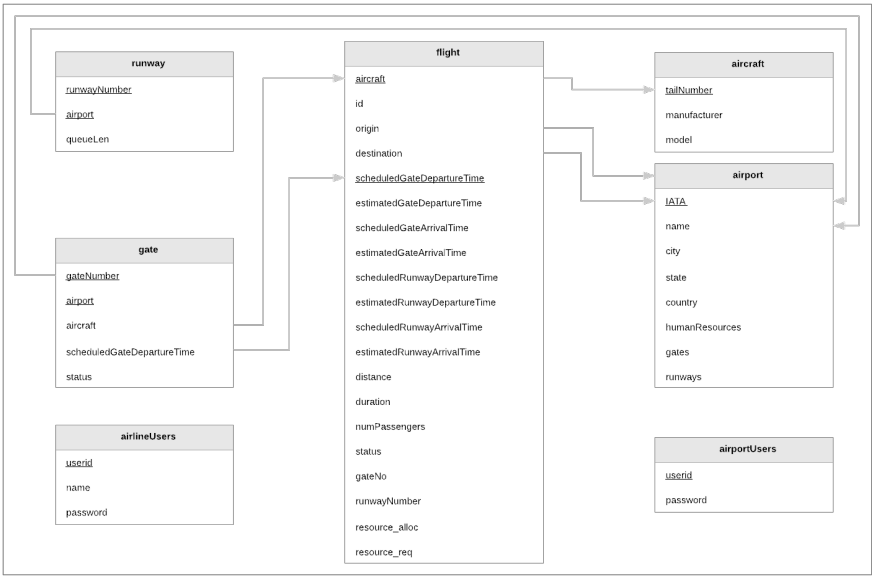

Figure 2: **Relational Schema** This figure shows the schema for the above Entity Relationship Diagram. It consists of seven relations and various foreign key constraints.