Airport Traffic Simulation II

CSE 6730 – Spring 2017 Project 2

Xun Cao, Xian Li, Jing Gu, Zhijie Xi, Shengji Ye
Georgia Institute of Technology
Atlanta, GA 30332, USA
xcao65, xli684, jgu47, zxie62, sye43@gatech.edu

***Abstract*—In this project, an airport simulation was implemented, which includes three more features than the previous project, multiple runway, emergency event handling and airway capacity control. The detail implementation for these three features will be described later and the simulation performance will also be evaluated.**

Keywords—Discrete Event, Airport Traffic, Queuing System

I. Introduction

The problem of airport congestion is receiving widespread attention.[1] Most major airports suffer from oversaturation of airspace and air-traffic facilities. Passenger-handling areas and ground-transport installations are equally congested during certain periods of time. In accordance with this view, the airport simulation was implemented to analyze this problem. An airport consists of a surface known as a runway for a plane to take off and land. The problem with the system is, the landing and departing aircraft compete with each other for one limited resource, the runway.[2] We would like to develop airport environment simulating the runways for both arrivals and departures. Our simulation goals include the main interaction of the landing and departing aircraft objects and the necessity to provide queues for departing and landing aircraft that cannot be given an immediate clearance from the airport controller, make it interesting to provide some statistics about the system that can serve in determining or measuring the performance of the airport for long periods. Beside, some other features such as multiple runways for each airport, emergency airport events and limited capacity of airlines will be offered so that the airport simulator is more diverse and close to real scenario.

In this project Java was employed, three features, multiple runways, limited capacity of airways and emergency events were implemented and the performance was evaluated to see the consistency of simulation results between distributive and non-distributive algorithms. What's more, before and after the features were inserted, the results of simulation were recorded to see if they affect the simulation results.

This report is designed to show the development of a discrete event (as opposed to a time-stepped) simulation that models a simple manufacturing system. This simulation is known as a queueing model in the literature. Queueing models are widely studied and used to model systems such as customers using a facility (e.g., a department store or restaurant), air and road transportation systems, computer networks and computer systems, and manufacturing systems, to mention a few. All of these systems involve customers (e.g., people, aircraft, vehicles, data packets, computer jobs, parts) that must travel from one station to another and receive service or processing at each station. Each station can only process one of these entities (e.g., a customer) at a time. Customers wanting to use a facility that is busy handling another customer must wait in a queue until the station is ready to process the next customer. [3]

The airport simulation model is one of the application for discrete event systems. Usually the landing and departing processes are monitored by air traffic controllers. The situation is more complex in real world. In the simulation of this report, the model is simplified as one runway per airport. The idea behind the system is the same as what happens in real life and the simplified model can give a good view of how this system works.

II. Problem Definition

Airport simulation is the computer-based modeling of any real-world process involved with an airport. Simulation allows organizations in the industry to analyze and experiment with their processes in a virtual setting, reducing the time and cost requirements associated with physical testing. Runways, gates and customer flow can be properly arranged and assessed within a simulation model, allowing companies to determine how best to fully utilize their resources at the lowest possible cost.

In this project, three features are added, and their detail description is offered below.

A. Multiple runways at each airport

Multiple runways were designed to represent the real situation in airport, however, in our simulation, there is only one airplane on a particular lane of multiple runways at the same time to avoid the conflicts and increase the safety.[4]

B. Emergency Airport Event

The emergency airport event and its handler was added to simulate the accidental events. In our implementation, when the emergency airport occurred, the source airport will be closed and the flight from some other airports to this one will be rescheduled and some airplanes about to land will be forced to landed to the nearby airports. [5]

C. Limited capacity of Airways

The feature of limited capacity of airways was created to deal with the air-traffic congestion. In our project, the airway capacity was set between airports randomly, if the airway traffic reaches the capacity currently, then the flights needs to be rescheduled until the airways between the two are available.[6]

III. Model Descriptions

Specifically, the system of interest is airport model with one runway. An airplane leaving one airport move flis to another airport. The runway of an airport can be either busy or free. If the runway is idle when the airplane arrives, it begins landing process. If the runway is busy, the airplane is placed in a queue where it will wait until the runway is free. Airplanes are removed from the queue in first-in-first-out (FIFO) order.[7]

An air traffic "controller", which is the simulator in the simulation model, is created to help schedule and maintain the future event list. The events of an airplane are scheduled based on runway condition and its current event type. This is controller by the simulator.

The simulation computes the total amount of circling time for each airport, which is the time where an airplane is ready to land but is waiting for an opening. The number of passengers arriving and departing each airport is provided as well. The simulator requires user-defined number of aircrafts, length of simulation in minutes and some random seeds. Other factors such as the aircraft speed and capacity can also be changed to user-defined, but for the sake of simplicity of this simulation, I keep these variables uniformed distributed and more detailed discussions can be found in next section.

IV. Model Design

The modeling of an airport traffic flow consists of four states in our simulation. They are arrival, landed, takeoff, and departure.

1) Arrival State

The arrival state of an airplane means there is a new aircraft arrives at the region of an airport. If the runway is free, it will begin to land, i.e., schedule an landed state event. Otherwise, the aircraft must circle around, which I add this aircraft to the queuing system.

2) Landed State

The landed state of an airplane means it has completed its landing at the airport. The number of arrival passengers will be added here. Then after a certain period of on-ground time, this aircraft will need to depart. So an occurrence of a landed state will automatically schedule a depart event.[8]

3) Take-off State

The depart state of an airplane means the aircraft finishes its ground service and is ready to takeoff. But it needs to check if the runway is free. It will begin to take off if runway is free otherwise be added to queuing system.

4) Departure State

The left state of an airplane means the aircraft is actually taking off from one airport to another airport. Then an arrival event will be scheduled automatically based on the distance between airports and speed of the aircraft. Also, the number of departure passengers will be added at this state.

B. Queuing System

The queuing system won't be used if the runway is always free. This won't happen in life. Then a queuing system needs to be introduced to schedule events if the runway is occupied. I use two queues, arrival queue and departure queue, to keep track of the events. There are three kinds of "deque" methods: Landing First, Departing First, and Time First.[9]

V. Implementation Detail

A .Multiple runways at each airport

In this simulation, a Boolean array named "runwayFree" was created as one of the fields of an airport object and the capacity of this array was initialized during the construction of this airport object. All the elements in the array were originally set true, which means all the runways are available. Four air events are related to this array, which are plane arrives, plane landed, plane take off and plane departs.

1) When a plane arrives, our simulation event handler will first check the status of the availability of runways, if at least one runway is available, then the plane can be arranged for a landed event, setting corresponding runway to false. if there is no available runway, then this event will be added to a queue to be rescheduled in the future.

2) When a plane was dealt with as a landed event, the runway that plane landed will be available for later planes. The simulation event handler will first check

the FIFO queue of the airport and deal with the first arrival event or takeoff event in the FIFO queue. If the FIFO queue is empty, set this runway to true.

3) For the event of take off, it is the same as the arrival event: when a plane is going to take off, the availability will be first checked, if there are at least one available runways, then the event can be arranged as a depart event, setting the corresponding runway to false. Else, this event will be added to the queue to be rearranged in the future.

For the last event, it's similar to the land event: when a depart event happens, the runway used before will be available for later planes. The simulation event handler will first check the FIFO queue of the airport and deal with the first arrival event or takeoff event in the FIFO queue. If the FIFO queue is empty, set this runway to true.

B. Emergency Airport Event

If an airport encounters emergency situation, it will be closed. Our implementation strategy for this is: When we have an emergency event(given in initialization), we will schedule two emergency events into the global event list: one with emergency event start timestamp (EEST), another with emergency event end timestamp (EEET).

The EEST event is to take care of airplanes which circle around the airports when the emergency event happens. The EEET event is to take care of the airplanes which will arrive during the emergency event period. Every time when an emergency event is popped out, it will query matched airport (with this emergency on) FIFO queue to pop out all the arrival events and schedule them for another destination(another airport near the airport the emergency event is on). If the emergency event is EEET, it will also popped out the departure events which are in the FIFO queue.

C. Limited capacity of Airways

In our project, a field named "airwayCapacity" and an array "m_airwayNumber" were created in the class Airport and the "airwayCapacity" field was initialized as 10, which means all the airway capacity between any airports is limited to 10. Besides, a method name "setAirwayNumber" was implemented to change the value of available airways according to the current flight.

Before scheduling a departure event, the event handler will check the number of flights on this plane's airline. If it is larger than the capacity, the plane will be pushed into the FIFO queue to wait for departure. The number of flights for each airline is maintained by an array in each airport. Each departure event will add one to its corresponding airline's flight number. And each arrival event will minus its corresponding airline's flight number by one.

VI. Results

After a certain period of user-defined simulation runs, the circling time of each airport and number of passengers arriving and departing each airport can be generated. Every experiment will run 6000(unit: minutes).

In this part, we will analyze our simulation results. By controlling the variable of the 3 features we added in project 2, we will draw comparison between the original version and the version with the specific feature added in to see what is the impact of every feature.

A. Number of Runways

By setting the number of runways of each airport to 1, 2, 4, 8 and setting the number of airplanes to 50, 200, 500, we record average cycling time of airports. Other settings include 10 airway capacity and emergency event on. And the results are shown as below:

FIGURE I. NUMBER OF RUNWAYS

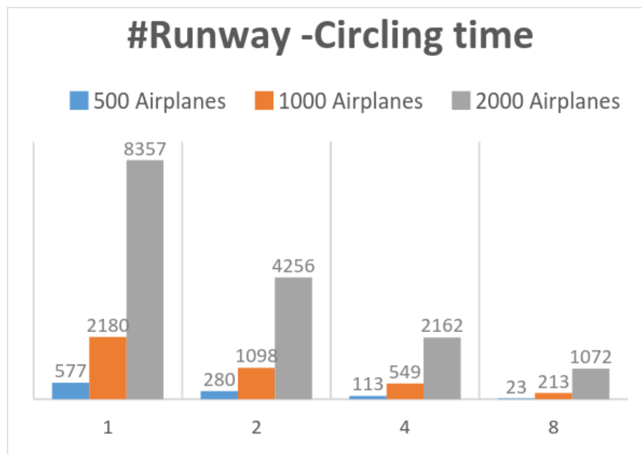


Figure I shows that in every experiment group with the same number of runways, more airplanes will lead to more cycling time while more runways will lead to less cycling time.

We can think about it empirically. When an airport only has 1 or 2 runways, it is highly possible that two flights will have time conflict in this airport which lead to delay of landing and high cycling time. When the number of runways increases, it will solve this problem. However, it is also related to number of airplanes. When the number of airplanes increases, the increase of runways might not be able to handles this. So the the number of runways should be able to handle the number of airplanes to reduce cycling time.

B. Emergency Event

By whether to schedule the emergency event and setting the number of airplanes to 50, 200, 500, we record average cycling time of airports. Other settings include 8 airway capacity and 2 runways on each airport. And the results are shown as below:

FIGURE II. EMERGENCY EVENT

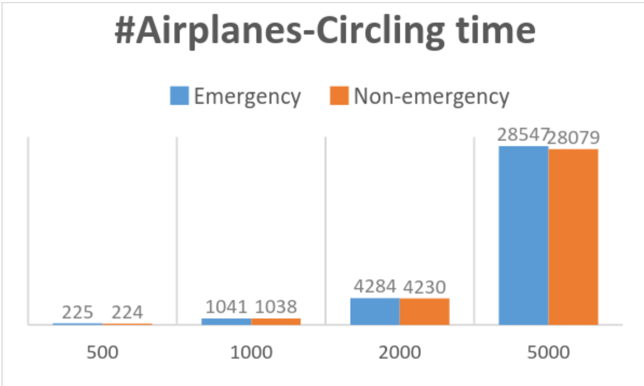


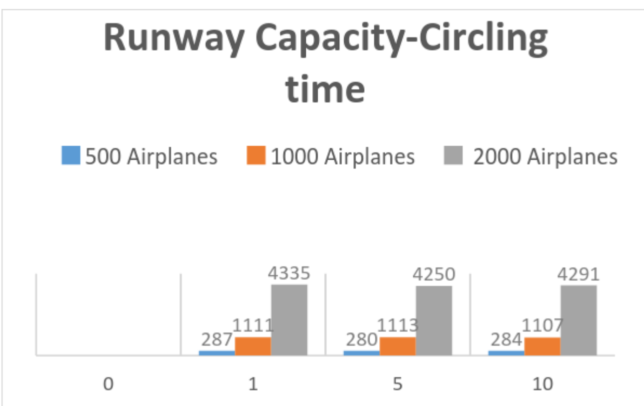
Figure II shows that with or without emergency event, the circling time will almost keep the same.

Regarding our emergency event logic, this result is in accordance with our implementation. When arrival event encounters emergency event, this plane will be redirected to other airports. So it has nothing to do with the circling time.

C. Airway Capacity

By setting the runway capacity to 0, 1, 5, 10 and setting the number of airplanes to 50, 200, 500, we record average cycling time of airports. Other settings include 2 runways on each airport and emergency event on. And the results are shown as below:

FIGURE III. AIRWAY CAPACITY



As shown in the figure III that if runway capacity is set to 0, the circling time will also be 0 for all cases. While 1, 5, 10 don't have too much difference.

It can be reasoned that when capacity is 0, no airplanes can take off. So the circling time for this will be 0. While the airway capacity will stop planes for taking off, so it has nothing to with the circling time if it is not 0. So the result is in accordance with the implementation.

VII. CONCLUSION

From the analysis above, it can be concluded that the number of airplanes and the number of runways impact the circling time together. And the number of airplanes is positively related to circling time, while the number of runways is negatively related to circling time. The emergency event has nothing to do with circling time and the airway capacity will only make a difference when jumping from 0 to positive. These results are generated based on my assumption and design of the simulation.

VIII. FUTURE STUDIES

This project simulates the very simple condition where the airport's locations are linearly distributed on the planet because it takes too much effort to load more than 50 airport's locations in our program. And some plain assumptions are made on the airplane types as well as air traffic process time. To make the simulation more related to the real world, it requires more complex simulation with more randomization on the parameters in the model. If time permitted, the model can be improved by using the real distribution of airplane landing, on-ground and departing time. And these time measures depend on the type of the airplane. If real-world airport location files are provided, it will improve the reality of our simulation.

In addition, many other factors might have influence of the simulation model. Weather for example, is a very important factor for airport operation. In our program, emergency event is pre-scheduled manually. But in reality, it is even harder to model the weather condition. In a word, a simple model can give the basic idea, but the accuracy and preciseness of the simulation requires lots of factors to be included in.

REFERENCES

[1] https://aspm.faa.gov/Default.asp
[2] https://www.quora.com/How-long-does-it-take-to-land-an-airliner-starting-from-cruising-altitude
[3] https://en.wikipedia.org/wiki/Cruise_(aeronautics)
[4] http://www.movable-type.co.uk/scripts/latlong.html
[5] https://www.world-airport-codes.com
[6] https://en.wikipedia.org/wiki/Discrete_event_simulation
[7] http://lyle.smu.edu/emis/cmmi5/Ibarra/DeskTop/PhD_Journals_Used/discrete_event_airline_lee_2003.pdf
[8] https://en.wikipedia.org/wiki/Traffic_simulation
[9] https://www.acm-sigsim-mskr.org/Courseware/Fujimoto/Slides/FujimotoSlides-04-ProcessOrientedSimulation.pdf