CSCE 1101-01, 02, 05 Spring 2023 Term Project Report

Simulating Randomness in Airport Landing Queue: Estimating Average Waiting Time and Analyzing Runway Occupancy

Authors

Mohamed Azouz 900225230

Hadi Hesham 900221206

Saif Abd Elfattah 900225535

Department of Computer Science and Engineering, AUC

Abstract:

This paper introduces an airport simulation queue model for analyzing runway occupancy and plane waiting times. The model considers factors such as arrival rate, landing time, and average time between arrivals. A pseudo code representation outlines the simulation methodology, while data analysis reveals the relationship between arrival rates, landing times, and runway occupancy. The findings provide insights for optimizing airport operations and minimizing delays. The simulation model serves as a valuable tool for enhancing efficiency and passenger experience. Future research can explore advanced optimization techniques for runway scheduling and capacity planning.

Keywords: Airport simulation, queue model, runway occupancy, plane waiting times, Optimization.

Introduction

Efficient airport operations are crucial for ensuring seamless air traffic flow and providing a safe and convenient travel experience for passengers. Among the key factors contributing to efficient airport management is the effective handling of landing queues, where airplanes patiently await their turn to touch down on a runway. Optimal queue management minimizes delays, optimizes resource utilization, and enhances overall airport performance.

The objective of this project is to develop a program that simulates the queueing operations of an airport, with a specific focus on estimating the average waiting time for airplanes in the landing queue before they can land. By conducting this simulation, valuable insights can be gained into the efficiency of the landing process, enabling the identification of potential areas for improvement and optimization.

By accurately estimating the average wait time in the landing queue, airport managers can assess the adequacy of their current systems and implement targeted measures to reduce congestion and enhance overall efficiency. Therefore, this simulation can serve as a valuable tool for airport authoritie***s.*** The program will replicate real-world scenarios, taking into account factors such as a single landing runway, random arrival times, and varying service times. The program will facilitate efficient management of the landing queue, allowing for seamless insertion and removal of airplanes as they await their turn to land.

Furthermore, the simulation will operate within a fixed simulation period, providing a comprehensive analysis of queue dynamics and enabling a meaningful evaluation of performance. The log of arrival and landing events will provide valuable information for analyzing the project's data. This data analysis will help identify any areas of congestion or problems and allow for the implementation of effective solutions to address them.

Problem Definition

The problem we aim to address in this project is to simulate the waiting queues of airplanes in an airport and calculate the average wait time in the landing queue. We will focus on a specific scenario of an airport with a single landing runway. When airplanes arrive near the airport, they are required to join a landing queue. Each plane has a random arrival time, denoted as Tarrival, which represents the time at which it arrives near the airport. Once in the landing queue, a plane may have to wait for a certain duration, Twait, until the runway becomes free and ready to receive it.

After a plane land on the beginning of the runway, the runway remains occupied for a fixed service time, Tlanding, until the plane successfully docks. During this service time, no other planes can land. To conduct the simulation, we will utilize a Double-ended Queue (DEQ) data structure. A DEQ allows for efficient insertion and removal of elements at both ends, which suits the requirements of our landing queue simulation. The program will be designed to run for a fixed simulation period, denoted as Tmax. We will consider the time unit as one minute for simplicity. Additionally, we will assume a random arrival time, Tarrival, for each plane, with a fixed average inter-arrival time, ΔT. Moreover, no plane will leave the queue until it successfully lands on the runway.

To enhance the simulation, the program will generate a log of the arrival and landing events, capturing the time of each event and the corresponding plane's information. This log will provide valuable insights into the sequence of events and can be used for analysis and further optimization. By varying parameters such as the inter-arrival time ΔT or the landing service time Tlanding, we can explore different scenarios, such as simulating peak and slack times of the day or assessing the impact of changes in landing duration. The C++ Random Number Generator (RNG) will be utilized to generate the time of arrival for each plane and determine whether it will land or not based on certain criteria.

Through this simulation project, we aim to gain a deeper understanding of the dynamics of landing queues at airports and provide insights into optimizing queue management for improved efficiency and passenger experience.

Methodology

To begin the simulation, the necessary libraries and header files are included in the code. The classes runway and plane are defined to represent the runway and individual planes, respectively. Additionally, the random number generation for the distribution is initialized to ensure randomness in the simulation. The simulation follows a loop that runs until the current time (t) reaches the maximum time (tMax) specified for the simulation. At each iteration of the loop, the current time is printed to the console to keep track of the progress.

During each iteration, a random number x is generated using the Random function. This random number represents the probability of a plane arriving at the airport. If the probability of a plane arrival, determined by run.getProb(), is less than x, a new plane is created. The arrival time of the plane is set to the current time, and it is added to the rear of the queue. A message is printed to indicate the arrival of the plane, providing information about its identification.

Next, the simulation checks if the queue is empty or not. If the queue is not empty, the simulation examines the plane at the front of the queue. If the arrival time of the plane at the front of the queue, plus the landing time, is less than or equal to the current time (t), it means the plane is ready to land. The simulation then checks if the runway is occupied or not. If the runway is not occupied, the plane is allowed to land. A message is printed to indicate the landing of the plane, including information about the plane's identification and the time it spent waiting in the air. The waiting time is updated accordingly. The plane is removed from the front of the queue, and the landed count is incremented. However, if the runway is occupied, a message is printed to inform that the runway is currently occupied, and the plane cannot land at the moment. If there are no planes in the queue, a message is printed to indicate that there are currently no planes waiting to land.

After processing the current time, the simulation increments the time (t) and proceeds to the next iteration of the loop. This ensures that the simulation progresses in chronological order. Once the simulation loop concludes, the average wait time for the planes that have landed is calculated. This is achieved by dividing the total waiting time by the number of planes that have successfully landed. The average wait time, along with the number of planes that have not yet landed, is printed to the console and written to an output file (excel).

In summary, the airport simulation queue project utilizes a C++ implementation to simulate the arrival and landing of planes at an airport runway. The simulation follows a methodology that involves initializing the necessary objects and variables, running a loop to simulate the passage of time, and processing plane arrivals and landings with a random function. The simulation outputs the average wait time for landed planes and keeps track of the number of planes that have not yet landed.

Specification of used Algorithms

Simulation Loop: -

The simulation loop iterates over a specified time range (tMax).

At each iteration, the current time (t) is printed to the console.

Random number x is generated using the PoissonRandom function to simulate plane arrivals.

If the probability of a plane arrival (run.getProb()) is less than x, a new plane is created, assigned an arrival time, and added to the queue.

Plane Landing: -

If the queue is not empty, the simulation checks whether the front plane's arrival time plus the landing time (landTime) is less than or equal to the current time (t).

If the runway is not occupied and the condition is satisfied, the plane is landed, and the waiting time is updated.

If the runway is occupied, a message is printed indicating that the plane is currently landing.

If the condition is not satisfied, a message is printed indicating that there are no planes currently landing.

Random Number Generation:

The RNF function is used to generate a random number x following a Poisson distribution.

The random number x is used to determine the probability of a plane arrival.

The Poisson distribution allows for the simulation of random arrivals based on a given average rate, inter-arrival time (avgTime).

Average Wait Time Calculation: -

After the simulation loop, the average wait time for the landed planes is calculated.

The total waiting time is divided by the number of planes that have successfully landed.

Relationships and Effects of Parameters:

Maximum Time (tMax) and Number of Planes: -

Increasing the maximum time (tMax) allows for a longer simulation duration, accommodating the potential arrival and landing of more planes. A longer simulation duration can result in a higher number of planes generated, as more time allows for more plane arrivals. The relationship between tMax and the number of planes generated depends on the probability of plane arrivals determined by our RNG.

Average Time between Arrivals (avgTime): -

The avgTime parameter affects the probability of a plane arrival.

A lower avgTime value increases the probability of a plane arriving at any given time, resulting in more frequent plane arrivals.

Conversely, a higher avgTime value decreases the probability of a plane arrival, resulting in less frequent plane arrivals.

Landing Time (landTime): -

The landTime parameter represents the time required for a plane to land.

A shorter landTime allows for faster plane turnover on the runway, reducing waiting times for subsequent planes.

Conversely, a longer landTime extends the time each plane spends on the runway, potentially increasing waiting times for other planes in the queue.

Pseudo code

```
// pseudo code
Initialize variables and objects
Open output file

while t < tMax:
    Print current time

    Generate random number x using PoissonRandom function

    if run.getProb() < x:
        Create a new plane
        Set arrival time for the plane
        Add the plane to the queue
        Print plane arrival message

    if the queue is not empty:
        if the front plane's arrival time + landTime <= t:
            if the runway is not occupied:
                Land the plane
                Print plane landing message
                Update waiting time
                Remove the plane from the front of the queue
                Increment landed count
            else:
                Print runway occupied message
        else:
            Print plane currently landing message
    else:
        Print no planes currently message

    Increment t

Calculate average wait time
Print average wait time and number of planes that haven not landed to console and output file

Close output file
```

Data Specifications

Time	Overall Status	Runway status	Percentage of runway occupancy
0:01	No planes currently	No	0
0:02	No planes currently	No	0
0:03	No planes currently	No	0
0:04	No planes currently	No	0
0:05	No planes currently	No	0
0:06	No planes currently	No	0
0:07	No planes currently	No	0
0:08	Plane 1 has arrived in the airspace & Plane 1 is landing and waited 0 minutes in the air	Yes	1
0:09	No planes currently	Yes	1
0:10	No planes currently	Yes	1
0:11	No planes currently	Yes	1
0:12	Plane 2 has arrived in the airspace	Yes	-1
0:13	Occupied	Yes	-1
0:14	Occupied	Yes	-1
0:15	Occupied	Yes	-1
0:16	Occupied	Yes	-1
0:17	Occupied	Yes then plane 1 left after service	1
0:18	Plane 2 is landing and waited 6 minutes in the air	Yes	1
0:19	No planes currently	Yes	1

0:20	No planes currently	Yes	-1
0:21	Plane 3 has arrived in the airspace	Yes	-1
0:22	Occupied	Yes	-1
0:23	Occupied	Yes	-1
0:24	Occupied	Yes	-1
0:25	Occupied	Yes	-1
0:26	Occupied	Yes	-1
0:27	Occupied	Yes then plane 2 left after service	1
0:28	Plane 3 is landing and waited 7 minutes in the air	Yes	1
0:29	No planes currently	Yes	1
0:30	No planes currently	Yes	1
0:31	No planes currently	Yes	1
0:32	No planes currently	Yes	1
0:33	No planes currently	Yes	1
0:34	No planes currently	Yes	1
0:35	No planes currently	Yes	1
0:36	No planes currently	Yes	1
0:37	No planes currently	Yes	1
0:38	No planes currently	Yes then plane 3 left after service	0
0:39	No planes currently	No	0
0:40	No planes currently	No	0
0:41	Plane 4 has arrived in the airspace & Plane 4 is landing and waited 0 minutes in the air	Yes	1
0:42	No planes currently	Yes	1

0:43	Plane 5 has arrived in the airspace	Yes	-1
0:44	Plane 6 has arrived in the airspace	Yes	-1
0:45	Occupied	Yes	-1
0:46	Occupied	Yes	-1
0:47	Occupied	Yes	-1
0:48	Occupied	Yes	-1
0:49	Plane 7 has arrived in the airspace	But now time ended and plane 3 didn't yet finished service	-1

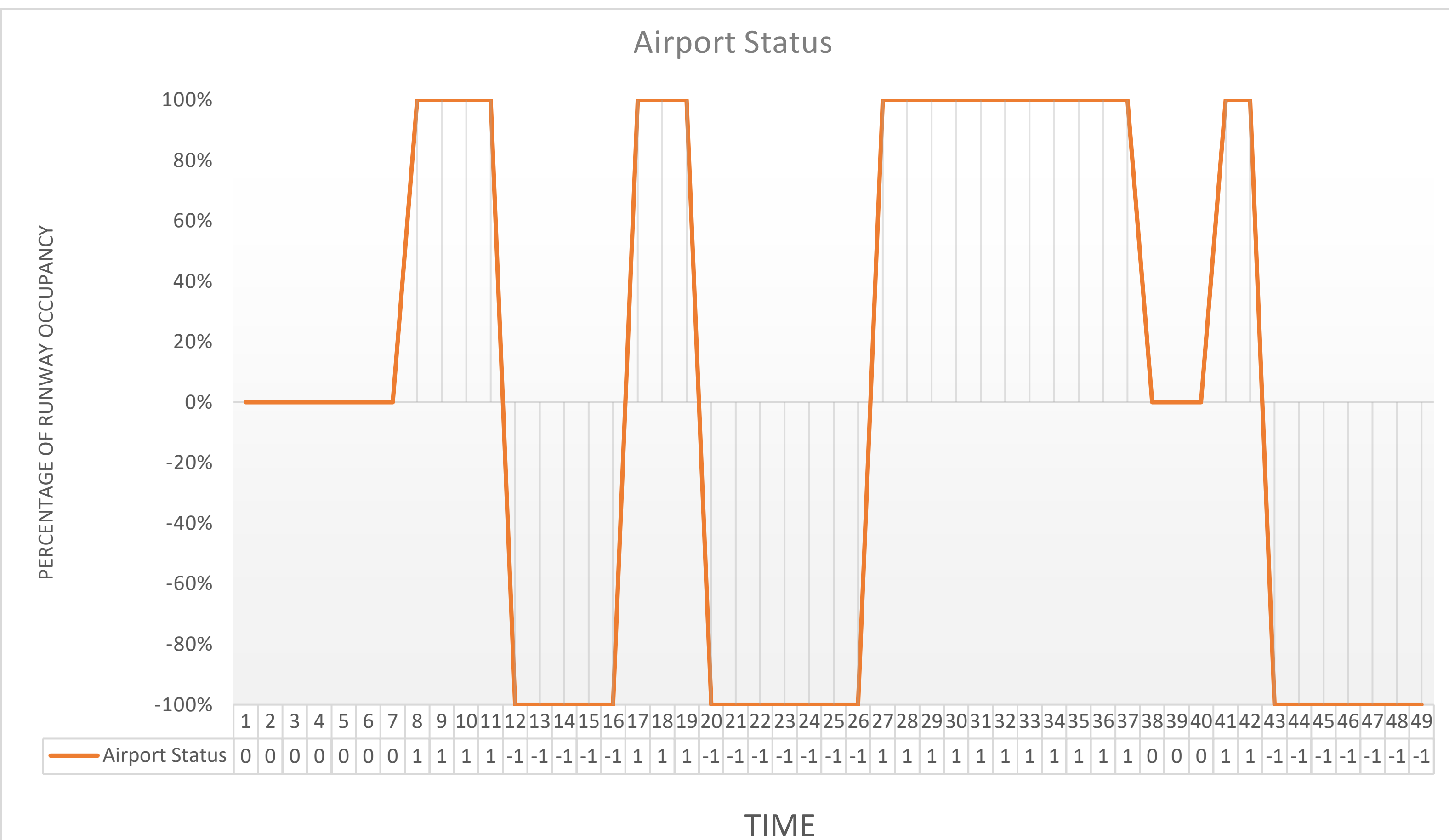

Figure 1: *Runway status in the Airport simulation*

Definitions: -

1 means that the runway is occupied but no other planes are waiting to land.

-1 means that the runway is occupied but the is at least one plane waiting to land.

0 means that the runway is not occupied and ready to receive planes.

Analysis of the data: -

Analyzing the provided data, we can observe the status of the simulation at different time intervals and the corresponding runway status and percentage of runway occupancy. Here are some key observations:

Time intervals 0:01 to 0:06:

The overall status is “no planes currently" throughout these intervals.

The runway status is not occupied).

The percentage of runway occupancy is 0%.

Time interval 0:08:

Plane 1 has arrived in the airspace and successfully landed, waiting 0 minutes in the air.

The runway is now occupied.

The percentage of runway occupancy is 100% (as there is only one runway and it is occupied).

Time intervals 0:09 to 0:11:

The overall status remains "No planes currently."

The runway status is "Yes" (occupied).

The percentage of runway occupancy remains at 100%.

Time interval 0:12:

Plane 2 has arrived in the airspace, but it cannot land immediately due to the occupied runway.

The runway status is still "Yes" (occupied).

Time intervals 0:13 to 0:16:

The runway remains occupied.

The percentage of runway occupancy remains at 100%.

Time interval 0:17:

Plane 1 has left the runway after service.

The runway status changes to "No" (not occupied).

The percentage of runway occupancy drops to 0%.

Time interval 0:18:

Plane 2 finally lands after waiting for 6 minutes in the air.

The runway is occupied again.

The percentage of runway occupancy increases to 100%.

Time intervals 0:19 to 0:26:

The runway remains occupied, and the percentage of runway occupancy remains at 1.

Time interval 0:41:

Plane 4 has arrived in the airspace and successfully landed, waiting 0 minutes in the air.

The runway is now occupied.

The percentage of runway occupancy is 1.

Time intervals 0:42 to 0:34

The runway remains occupied, and the percentage of runway occupancy remains at 1.

Time interval 0:43:

Plane 5 has arrived in the airspace, but it cannot land immediately due to the occupied runway.

The runway status is still occupied.

Time intervals 0:45 to 0:48:

The runway remains occupied.

The percentage of runway occupancy remains at 100%.

Time interval 0:49:

Plane 7 has arrived in the airspace, but the simulation time has ended, and plane 3 has not yet finished its service.

The runway status remains occupied.

Results: -

The average wait time for the 4 planes which landed is: 13 minutes

Number of planes didn't land: 3

The analysis of the data shows the dynamics of the airport simulation queue. The runway's occupancy status is influenced by the arrival of planes and the availability of the runway. When the runway is occupied, planes have to wait in the air until it becomes available. The percentage of runway occupancy indicates the utilization of the runway during different time intervals.

The data demonstrates that the runway occupancy is directly related to the arrival rate of planes, the time required for a plane to land (landTime), and the average time between plane arrivals (avgTime). When the arrival rate of planes is higher or the landTime is longer, the runway tends to remain occupied for longer durations, resulting in a higher percentage of runway occupancy. Conversely, when the arrival rate is lower or the time landing is shorter, the runway occupancy becomes zero or negative one (decreases).

Conclusions

In conclusion, the airport simulation queue model presented in this paper provides a valuable tool for analyzing and optimizing runway occupancy and plane waiting times. By considering factors such as arrival rate and landing time, the model offers insights into the dynamics of airport operations. The data analysis reveals the relationship between these factors and runway occupancy, highlighting the impact on efficiency and delays. The findings emphasize the importance of effective runway scheduling and capacity planning to enhance overall performance and passenger experience. The simulation model can assist airport authorities in making informed decisions for improving operations and minimizing delays. Future research can further explore advanced optimization techniques to refine runway scheduling strategies.

Acknowledgements

I would like to extend my heartfelt appreciation to Dr. Howida, my instructor, for her invaluable support and mentorship during the research journey. Her extensive knowledge and valuable perspectives on the queue simulation have greatly influenced my comprehension of the subject matter. Additionally, I am grateful to Mr. Mohamed Basuony, my teaching assistant, for his insightful explanations on the project's framework. His assistance has significantly enhanced the quality of this research paper. Thanks to my friend Kerolos for helping us overall. Lastly, I would like to express my gratitude to all the authors whose works have been referenced in this paper, as their contributions have enriched the content and depth of my research.

References

GeeksforGeeks. (2023a). Implementation of Deque using doubly linked list. *GeeksforGeeks*. https://www.geeksforgeeks.org/implementation-deque-using-doubly-linked-list/

Output file into Excel line by line. (n.d.). Stack Overflow. https://stackoverflow.com/questions/33091841/output-file-into-excel-line-by-line

Zhao, N., & Zhang, J. (2022). Research on the Prediction of Aircraft Landing Distance. *Mathematical Problems in Engineering*, *2022*, 1-10.