Airport Traffic Analysis

– Modelling and Simulation Project

Paras Kaushik (2017UCO1514)

Akhil Vaid (2017UCO1521)

Dalip Patra (2017UCO1529)

Vaibhav Sharma (2017UCO1533)

COE-1

Netaji Subhas University of Technology, New Delhi

Introduction To The Problem

Airport facilities constantly face challenges due to the substantial increase in air traffic flow, costs associated with the fuel consumption and delays that aeroplanes must endure. In order to aid tower-control to best utilize the available resources, application of modeling and simulation to the airport operation was used to evaluate the efficiency of current strategy and predict the result of any future changes that might be applied to the system.

Theory behind the problem

What is simulation?

Simulation of a system is the computer based operation of a model, it makes it possible to study system properties, operating characteristics, evaluate different operational strategies, draw conclusions and make action decisions based on the results of the simulation before the actual implementation of the strategy is performed. It is a useful tool that guides improvement of existing resource utilization strategy and assist in planning and designing new strategies based on data gathering and information technology.

What are Decision Analysis Models?

a) **DECISION TREES**

Many decision analysis models are available. The most basic form is the use of decision trees which are diagrams of all the possible scenarios a patient may go through and expected outcomes of each one at the end of the branch. The tree branches model all possible outcomes with their associated probabilities. Decision trees work well when outcome set is small and defined with a short time horizon.

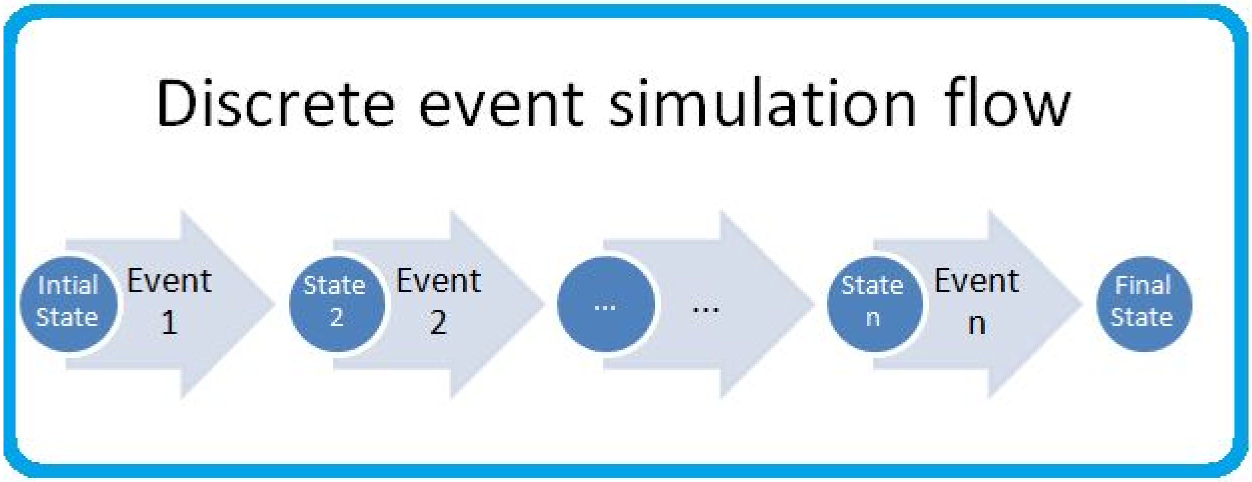
b) **MARKOV MODELS**

Markov models are cohort-based models on which a hypothetical homogenous cohort of patients moves through defined states. Although cohort-based models are relatively more simple, there are situations when individual-level outcomes must be represented. Moreover, markov models depends on the **"Markovian assumption"**, which assumes that transition probabilities do not depend on history, which makes markov models "memoryless". This can be addressed by increasing the number of states. If the number of required states grows too large, individual simulation is a better alternative. It also allows better representation of heterogeneity in characteristics.

One approach to implement individual-level simulations is Discrete Event Simulation (DES), which is an adaptation of methods borrowed from engineering and operations research. DES represents the problem in defined states and events that can happen to individuals that transfer them from one state to another and consequences of those events.

Background

What is Discrete Event Simulation?



DES is the simulation technique of choice when:

- Model has very large numbers of states
- Competition for resources and queues formation need to be represented in the model to help improve service waiting times and answer resource allocation questions.
- There is a need to represent interactions between individuals in the model

Discrete Event Simulation models a system in order to compare different strategies and identify the one that best utilize the system under investigation. The core concepts of DES are **entities, attributes, events, resources, queues and time.** Entities are objects of interest that have attributes which influence the probability of experiencing certain events. They compete for resources, and enter queues, over time.

Events may be an airplane entering the system or departing the airport. Queues formed by airplanes waiting for clear runway, are competing for resources which can be represented by various service disciplines such as First in First Out or a Priority based discipline according to the discipline used in the system under investigation.

Discrete time handling nature of DES means that simulation moves forward in time at “discrete” points that events occur at. The model advance to next event, without unnecessary interim computations, which makes the DES computation much faster.

Stages of DES modeling process:

The first step in developing DES is to define the system under investigation and relevant events that can occur to individuals in that system. This is assisted by performing a model structure in which all possible scenarios and their required parameters are identified. The next step is to estimate those parameters using pre-collected data, if required data are not available or of poor quality the expert in the system operation may be consulted for data elicitation and their responses should be validated. If expert elicited data are not of appropriate validity, the model should serve only as a guide for further data collection needed to build the model.

Model implementation involves transferring conceptual model structure into a computer program. Model can be populated and analyzed using DES dedicated software which may offer a better visualization of the problem through animations and also ease of application by a non-expert. Another approach is the use of a general programming language such as **"R".**

General programing languages are more flexible in representing details of the system, provide shorter simulation running times and less dependence on commercial software. However, they don't offer fancy animations and involve writing more code for package functions (such as: codes to formulate event list, queues formation, resource utilization and sampling from probability distributions) which makes its use more complex and requires extensive code debugging.

Model outputs may include mean values or distributions of values. Best practice recommends that the stability of output is considered acceptable if there is less than a 5% (or 1% according to the setting's threshold) difference between output values across model runs. Stability can be improved by running more entities, increasing time horizon or run more replications in one model according to the nature of the system being simulated.

Validation techniques should be applied to ensure that the model accurately represents reality. Sensitivity analysis should be carried out to optimize the simulation model and identify variables which have significant impact on the model.

In **DES reporting,** both general and detailed representations of a DES structure and logic should be provided along with detailed event documentation figures as they are essential to the modeler when presenting the model methodology and results.

Work Done by members of the Team

All members contributed towards the development of this project, specifically in the following areas:

1. **Paras**: Wrote the code for simulation and did Result Analysis.
2. **Akhil**: Compiled and wrote the report.
3. **Dalip**: Acquired the data for project and report.
4. **Vaibhav**:Data cleansing and debugged the code.

Approach

The problem was to simulate air traffic arriving and departing at the airport according to real world data.

We used the 'simmer' library for R, which provided many useful functions for the pipelining of functions to simulate appropriately.

The key terms in the project are:

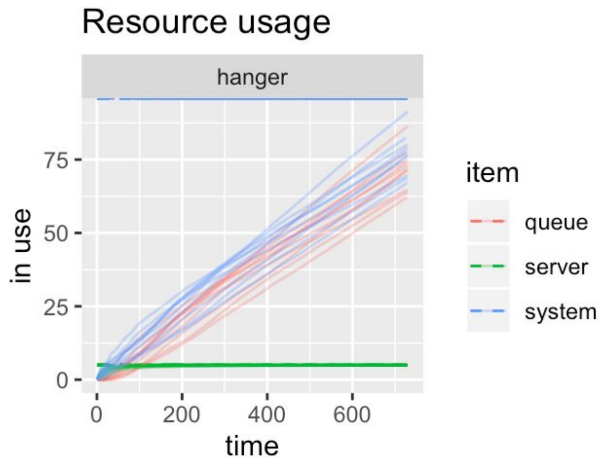
1. **Hangar** : the area where planes are stored after arriving at the airport. Here maintenance is done over the course of time, to make plane ready for the air again.
2. **Arrival time**: When a new airplane enters system, will try to search for a place in hangar.
3. **Departure time**: When an existing plane leaves system, reducing count of hangar by 1.

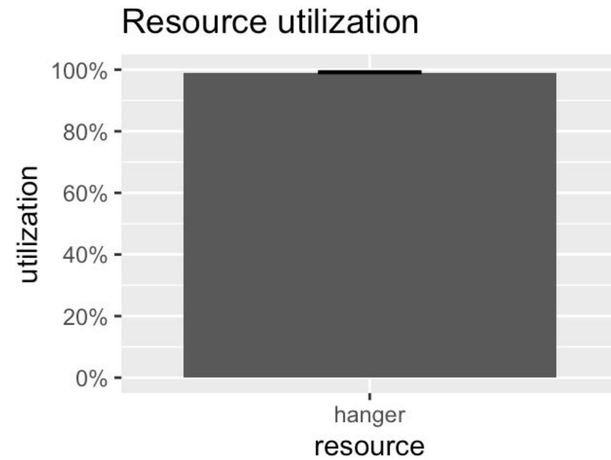
Why we chose this problem?

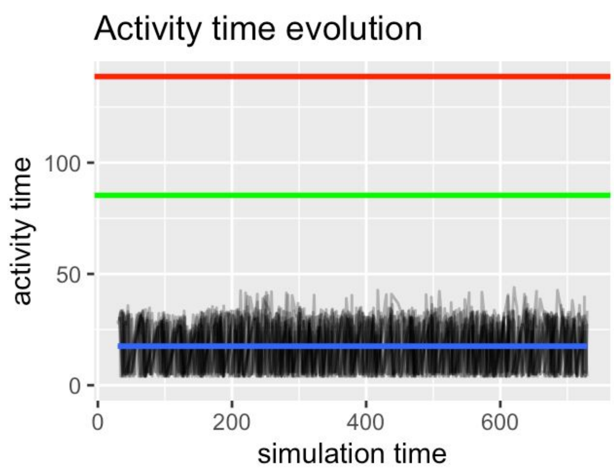
There are over 40,000 **airports** with codes given by the International Civil Aviation Organization (ICAO), a United Nations sponsored group. Every day, these airports have to work and handle thousands of aircrafts entering, arriving, departing, waiting at their airport. This is not an easy task and this can be modelled as a discrete event simulator.

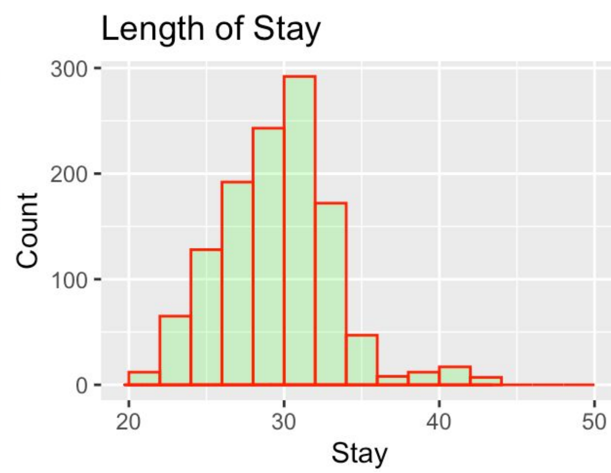
Also, we all have an interest in how airports can handle the large volume of traffic they receive each day and so took up this task.

Results analysis and Conclusion









PIC 1 (Resource Usage):-

This is showing the usage of the resource “hanger” w.r.t time by 3 components namely queue, server and system. The red line depicts the usage by queue, the green line depicts the usage by server and the blue line depicts the usage by system.

1) As the red line(queue) is increasing, we can as time passes, more and more planes will come and hence, the demand of the hangers will be more. So, it’s usage by hangers is increasing over time.
2) As the green line(server) is constant, we can see that the server demands a constant usage of hanger over time.
3) As the blue line(system) is increasing, we can see that the demand of the resource “hanger” is increasing over time by the system.

PIC 2 (Resource Utilization):-

This is showing the utilization of the resources. In this case, we have only one resource which is “hanger”.

By the graph, we can see that the hanger is occupied 100%, which is understandable considering its limit.

PIC 3 (Activity Time Evolution):-

This is showing the activity time as a function of simulation time. From the graph, we can see that it is constant over the whole simulation time.

PIC 4 (Length of Stay):-

This graph shows how many flights (Y-axis) are staying for a particular limit of time (X-axis) in minutes

We can see it as almost a normal distribution with decreasing count on the RHS of the mean.

Summary:-

> summary(n_arrivals$waiting)

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   4.667  11.908  81.469  24.966 607.779
```

> summary(arrivals_no_resource$waiting)

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00   43.65  135.70  261.08  563.87
```

> summary(arrivals_no_resource$ALOS)

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00   69.57   73.36  135.73  187.79
```

> summary(n_arrivals$activity_time)

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.43   26.90   29.38   29.39   31.59   46.61
```

> **summary(n_resources$queue)**

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   36.00   71.00   71.53  107.00  173.00
```

> summary(mydata$newflights)

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  170.0   189.0   199.5   199.9   209.2   230.0
```

```
> end_time = Sys.time()
> end_time - start_time
Time difference of 1.134027 mins
```

We note that the **time required for 100 repetitions of the simulation was 1.134 minutes.**

Conclusion

We can successfully see how we modelled and simulated air traffic at the airport. This inspires us to work on new and better solutions for Discrete event simulation and applying them in real life scenarios.

References

1. Law and Ketton book on Simulation
    a. https://fac.ksu.edu.sa/sites/default/files/index.pdf
2. https://support.rstudio.com/
3. https://www.r-bloggers.com/%F0%9F%96%8A-r-coding-style-guide/
4. https://www.researchgate.net/publication/269217652_Discrete-Event_Simulation_of_Air_Traffic_Flow
5. Data from Los Angeles International Airport Data.
    a. https://www.kaggle.com/cityofLA/los-angeles-international-airport-data