MAS547 - Group Project 3

Professor Maria Hulett

Group Member:

Wu, Ke

Zhou, Yingying

Chai, Rui

Wang, Xiaxin

November 30, 2020

Contents

[1. Model establishment 2](#_Toc57387655)

[1) Model overview 2](#_Toc57387656)

[2) Source, separator, and combiner setup 2](#_Toc57387657)

[3) Server Property Setup 7](#_Toc57387658)

[4) Path Weight 11](#_Toc57387659)

[5) Total Waiting Time, Total Cost, Average Waiting Time Set Up Process 11](#_Toc57387660)

[6) Pie Chart and Plot 20](#_Toc57387661)

[2. Theoretical result and experimental result comparison 22](#_Toc57387662)

[3. Experiment 24](#_Toc57387663)

[4. Simulation result Analysis 26](#_Toc57387664)

[5. Recommendations, Benefits, and obstacles 27](#_Toc57387665)

1. Model establishment
2. Model overview

There are 2 entities, 1 source, 5 servers, 1 separator, 1 combiner and 1 sink in the model. They are named Passenger, Baggage, Arrivals, Dropoff, WTMD, WTMD2, WTMD Check, Baggage Check, ETD, Baggage Pickup and Dep. Below is a figure of the model.

A picture containing graphical user interface, diagram

Description automatically generated

1. Source, separator, and combiner setup

In the Arrivals property, we set the entity type as Passenger and the interarrival time as Random.Exponential(0.2) based on the dataset.

Before setting up separator logic, we need to set two real state variables r in the ModelEntity. We name them as NumPassengers and NumBaggage as shown below.

Graphical user interface, text, application, email

Description automatically generated

Then we go back to the separator setup. A few steps need to take to complete the logic.

Step 1, since we have two entities, passenger and baggage, and a certain number of baggage need to be separated from passengers at drop-off station. Some people carry one baggage. Some carry more than one baggage. So, we set the separation mode as Make Copies and set copy quantity as ModelEntity.NumBaggage. Then set the copy entity type as Baggage.

Step 2, in the process logic, since we build the model to find out the optimal number of servers and workers in the system, we set the initial capacity as a new referenced property so that it will appear as Control in the experiment that we conduct later. Then, set processing time as Random.Gamma( 1.3664, 33.575) and time unit as seconds according to the dataset.

Step 3, according to the data, 41% of passengers carry only one baggage. 40% of passengers carry two baggage. 19% of passengers carry 3 baggage. So, we can set the baggage number distribution as Random.Discrete (1,0.41,2,0.81,3,1). Also, since we need to track total number of passengers in the system, we set a new real state variable under definition tab called totalpassengers. After setting up these steps, we connect the variables with the model through setting up On Entering in State Assignment. First, set ModelEntity.NumBaggage new value as Random.Discrete (1,0.41,2,0.81,3,1). Second, set totalpassengers new value as totalpassengers + 1. In this way, totalpasengers will record the total number of passengers by adding 1 to the variable every time where there is a passenger enters the separator. Third, we assign totalpassengers as ModelEntity.NumPassengers.

Step 4, we set workers as drop-off officers to supervise the drop-off process. Also, there is a drop-off officer station where officer must walk to there to work. There is also a drop-off home where is the home for drop-off officer. In the Secondary Resources, we set the object type as Specific, object name as dropoffofficer, choose the smallest distance as selection goal, and we request the worker to go to dropoffstation node.

Step 5, we set the desired speed of worker is 0.2 and initial node is dropoffhome. In the idle condition, they will park at node and wait until the next passenger. Since we need to find out the optimal number of officers in the system, we set the Initial Number in System as a new referenced property so that we can control the number of officers in the system in the experiment.

Step 6, considering real life situation, baggage drop-off is just a place that worker can work. It is the worker who really server the passenger. So, the capacity of separator, which is baggage drop-off, should be equal to the number of drop-off officers. So, we set the Initial Capacity as existing referenced property, which is dropoffofficer\_InitialNumberInSyste.

Those are all properties we need to set up in the separator except Add-on Process, which will be discussed later.

The set up are shown as below:

Graphical user interface, table

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Regarding the set up in combiner, also a few steps need to be taken.

Step1, we need to batch passenger and baggage. So, in the batch logic, we set Baggage Quantity as ModelEntity.NumBaggage, Matching Rule as Match Members and Parent.

Step2, in the process logic, we want to find out the optimal capacity. So, set the initial capacity as new referenced property so that we can control the capacity in the experiment. Then, set processing time Random.Lognormal( 3.4598, 0.99779) and time unit as seconds.

Those are all properties we need to set up in the combiner except Add-on Process, which will be discussed later.

The setup is shown as below.

Table

Description automatically generated

1. Server Property Setup

For WTMD server, in order to control the capacity in the experiment, we set the initial capacity as a new referenced property, which is called WTMD\_InitialCapacity. The processing time distribution is Random.Gamma( 1.1108, 43.606) as time unit is seconds according to dataset.

For WTMD2 server, in order to control the capacity in the experiment, we set the initial capacity as a new referenced property, which is called WTMD2\_InitialCapacity. The processing time distribution is Random.Lognormal( 1.667, 1.4892) as time unit is seconds according to dataset.

For BaggageCheck server, in order to control the capacity in the experiment, we set the initial capacity as a new referenced property, which is called BaggageCheck\_InitialCapacity. The processing time distribution is Random.Gamma( 1.1886, 41.761) as time unit is seconds according to dataset.

For WTMDCheck server, we set the worker, patdown officer, to work there. And we set a basic node as patdown officer home station and another basic node as patdown station. For patdown officer, we set the desired speed 0.2. Initial node is patdownofficerhome. When it’s idle, the officer will park at the patdown station and wait until the next passenger. Then set the Initial Number in System as a new referenced property, which is called patdownofficer\_InitialNumberInSystem. So, the initial capacity of WTMDCheck server is also referenced as patdownofficer\_InitialNumberInSystem. Then, set the processing time as Random.Lognormal( 2.0259, 0.91) and time unit as seconds. Last, in the secondary resource property, add a new row in Resources for Processing. Set specific as Object Type. Patdownofficer as object name. Set smallest distance as selection goal. Then set patdownstation as the destination node.

For ETD server, we set the worker, etd officer, to work there. And we set a basic node as etd officer home station and another basic node as etd station. For etd officer, we set the desired speed 0.2. Initial node is etdofficerhome. When it’s idle, the officer will park at the etd station and wait until the next passenger. Then set the Initial Number in System as a new referenced property, which is called etdofficer\_InitialNumberInSystem. So, the initial capacity of ETD server is also referenced as etdofficer\_InitialNumberInSystem. Then, set the processing time as Random.Lognormal( 1.0808, 1.1593) and time unit as seconds. Last, in the secondary resource property, add a new row in Resources for Processing. Set specific as Object Type. etdofficer as object name. Set smallest distance as selection goal. Then set etdstation as the destination node.

All the servers’ properties are done after the steps above expect for Add-on Process, which will be discussed later.

All properties are shown as below:

Graphical user interface, application, table

Description automatically generatedGraphical user interface, application, table

Description automatically generatedGraphical user interface, table

Description automatically generatedGraphical user interface, application, table

Description automatically generatedGraphical user interface, application, table

Description automatically generatedTable

Description automatically generated

Graphical user interface, application, table

Description automatically generatedGraphical user interface, text, application

Description automatically generatedTable

Description automatically generated

1. Path Weight

According to the data, 10.2% of passengers go to WTMD2 after WTMD, 3.3% of passengers go to WTMDCheck after WTMD, 77.2% of passengers go to BaggagePickup after WTMD, 9.3% of passengers go to ETD after WTMD, 9.1% of passengers go to WTMDCheck after WTMD2, 22.7% of passengers go to ETD after WTMD2, 68.2% of passengers go to BaggagePickup after WTMD2. So, we set the weight correspondingly.

1. Total Waiting Time, Total Cost, Average Waiting Time Set Up Process

Since Simio does not record total waiting time, so, we create a few variables and a few processes to record them.

First, we create 3 real state variables in States under Definition. We name then totalwait, waitingtime, and tmp.

Second, set up the Add-on Process Triggers in Dropoff separator. Double click on Processing and add an Assign step. Set the state variable name as waiting time, which we just created under definition tab. Then assign a new value, which is TimeNow – Entity.TimeCreated. This part returns the amount of time that the entity has been waited before getting served in Dropoff. Then double click on After Processing and add an Assign step. Set state variable name as tmp and assign a new value, TimeNow to it. This value just an intermediary value which will used later.

Third, set up the Add-on Process Triggers in WTMD. Double click on Processing and add an Assign step. Set the state variable name as waiting time, which we just created under definition tab. Then assign a new value, which is waitingtime + TimeNow – tmp. This part returns the amount of time that the entity has been waited before getting served in WTMD. Then double click on After Processing and add an Assign step. Set state variable name as tmp and assign a new value, TimeNow to it. This value just an intermediary value which will used later.

Forth, set up the Add-on Process Triggers in WTMD2. Double click on Processing and add an Assign step. Set the state variable name as waiting time. Then assign a new value, which is waitingtime + TimeNow – tmp. This part returns the amount of time that the entity has been waited before getting served in WTMD2. Then double click on After Processing and add an Assign step. Set state variable name as tmp and assign a new value, TimeNow to it.

Fifth, set up the Add-on Process Triggers in WTMDCheck. Double click on Processing and add an Assign step. Set the state variable name as waiting time. Then assign a new value, which is waitingtime + TimeNow – tmp. This part returns the amount of time that the entity has been waited before getting served in WTMDCheck. Then double click on After Processing and add an Assign step. Set state variable name as tmp and assign a new value, TimeNow to it.

Sixth, set up the Add-on Process Triggers in ETD. Double click on Processing and add an Assign step. Set the state variable name as waiting time. Then assign a new value, which is waitingtime + TimeNow – tmp. This part returns the amount of time that the entity has been waited before getting served in ETD. Then double click on After Processing and add an Assign step. Set state variable name as tmp and assign a new value, TimeNow to it.

Seventh, set up the Add-on Process Triggers in BaggageCheck. Double click on Processing and add an Assign step. Set the state variable name as waiting time. Then assign a new value, which is waitingtime + TimeNow – tmp. This part returns the amount of time that the entity has been waited before getting served in BaggageCheck. Then double click on After Processing and add an Assign step. Set state variable name as tmp and assign a new value, TimeNow to it.

Eighth, set up the Add-on Process Triggers in BaggagePickup. Double click on Processing and add an Assign step. Set the state variable name as waiting time. Then assign a new value, which is waitingtime + TimeNow – tmp. This part returns the amount of time that the entity has been waited before getting served in BaggagePickup. Then double click on After Processing and add an Assign step. Set state variable name as tmp and assign a new value, TimeNow to it.

Ninth, set up the Add-on Process Triggers in Dep. Double click on Entered and add an Assign step. Set the state variable name as totalwait. Then assign a new value, which is totalwait + waitingtime. This part returns the total amount of time that the entity has been waited.

Finally, after setting up the above, we need to create output statistics so that we can see the result later when we run the experiment. So, we create an output statistic called totalwaitingtime and make it equal to totalwait. Set the unit type as time and units as Minutes. In this way, we can record the total waiting time and make it visible in the result.

Below are pictures of all the setup above:

First, the state variables we created.

A picture containing graphical user interface

Description automatically generatedSet up in Dropoff Add-On Process Triggers.

A picture containing timeline

Description automatically generatedA picture containing graphical user interface, table

Description automatically generatedTable

Description automatically generated

Set up in WTMD Add-On Process Triggers.

A picture containing clock, meter

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Set up in WTMD2 Add-On Process Triggers.

A picture containing graphical user interface, Teams

Description automatically generatedA picture containing table

Description automatically generatedGraphical user interface, table

Description automatically generated

Set up in WTMDCheck Add-On Process Triggers.

A picture containing Teams

Description automatically generatedGraphical user interface, table

Description automatically generatedGraphical user interface, table

Description automatically generated

Set up in ETD Add-On Process Triggers.

A picture containing chart

Description automatically generatedGraphical user interface, table

Description automatically generatedGraphical user interface, application, table

Description automatically generated

Set up in BaggageCheck Add-On Process Triggers.

A picture containing chart

Description automatically generatedGraphical user interface, table

Description automatically generatedGraphical user interface, application, table

Description automatically generated

Set up in BaggagePickup Add-On Process Triggers.

A picture containing text

Description automatically generatedTable

Description automatically generatedGraphical user interface, table

Description automatically generated

Set up in Dep Add-On Process Triggers.

Graphical user interface, application, Teams

Description automatically generatedTable

Description automatically generated

Next, move to the average waiting time setup. Create an output statistics and name it as avgwaitingtime. Set the value of it as totalwait/totalpassengers. Unit Type is Time and units is Minutes. Next, we work on the total number of passengers. We also want to track the total number of passengers. So, we create an output statistics and name it as NumberofPassengers. Set the value of it as totalpassengers.

The setup is shown below:

Table

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Last, we want to know how much it will spend in an 8-day experiment. So, we create an output statistics and name it as totalcost. The costs involved here is the officer salary, WTMD machine cost and Baggage X-ray machine cost.

After looking up the data online, we find out the average salary per hour for officer at the airport is $13.02. So, the cost of increasing one officer in an 8-day experiment is 8\*24\*13.02 = 2499.84. We round it to 2500.

Then the price of a Rapiscan Metor 6E High Securoty Walk-Through Metal Detector is $6995. The service life period is 8 years. So, 6995/8/365 = 2.39. That means the cost of WTMD machine per day is $2.39. So, in an 8-day experiment, the cost would be 2.39\*8 = 19.16. We round it to $19.

Then the price of a Rapiscan 620XRve X-Ray Scanner machine is $21440. The service life period is 8 years. So, 21440/8/365 = 7.34. That means the cost of baggage X-ray machine per day is $7.34. So, in an 8-day experiment, the cost would be 7.34\*8 = 58.73.

So, the expression of the total cost would be

2500\*etdofficer\_InitialNumberInSystem+2500\*patdownofficer\_InitialNumberInSystem+2500\*dropoffofficer\_InitialNumberInSystem+19\*WTMD\_InitialCapacity+58.73\*BaggageCheck\_InitialCapacity+19\*WTMD2\_InitialCapacity.

Below are the setup and price of machines which we choose:

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface

Description automatically generated

1. Pie Chart and Plot

We have two response variables in the experiment, which are total cost and total waiting time. Since the cost is fixed in one scenario, we decide only to plot total waiting time. So, we create a status plot, keep track of totalwait and label it as TotalWaitingTime. The property is shown below.

Graphical user interface, application, Teams

Description automatically generated

This is how plot looks like after running a few hours.

Chart, bubble chart

Description automatically generated

In the pie chart, we want to keep track of the utilization of each server or officer or separator or combiner. For example, in the BaggagePickup, we set the expression as BaggagePickup.Capacity.ScheduledUtilization and label it as BaggagePickupUtilization. It’s similar for WTMD, WTMD2, WTMDCheck, ETD, dropoff, BaggageCheck. So, the property of pie chart is shown below.

Graphical user interface, application

Description automatically generated

This is how pie chart looks like after running for a few hours.

Chart, pie chart

Description automatically generated

1. Theoretical result and experimental result comparison

In order to make sure the model is representing the system to be studied, we conduct an experiment and make a comparison between simulated and theoretical results.

The experiment’s property we conduct is as follows:

1. Run length: 8 days, Warm-up period: 1 day
2. Number of Replication: 6
3. Capacity of WTMD, WTMD2, Baggage: 5
4. Number of pat down officer, etd officer, dropoffofficer: 5

Then we get the simulated utilization for each server and collect the data into a table.

Then we calculate the theoretical utilization. The results are shown below:

Table

Description automatically generatedTable

Description automatically generated

Table

Description automatically generatedTable

Description automatically generated

Table

Description automatically generatedTable

Description automatically generated

The comparison is as below:

|  |  |  |
| --- | --- | --- |
|  | Simulated | Theoretical |
| DropOffofficer | 0.7669 | 0.7667 |
| ETDofficer | 0.01364 | 0.013551 |
| Patdownofficer | 0.00815 | 0.00845 |
| WTMD | 0.8119 | 0.8 |
| WTMD2 | 0.0278 | 0.0238 |
| BaggageCheck | 1 | 1.479 |

As the result, we can see that the simulated values and theoretical values are very close except for the BaggageCheck. But both the simulated utilization and theoretical utilization indicate that the system cannot handle the passengers, which means the system will crush.

Table

Description automatically generated

Based on the comparison above, the model is representing the system to be studied. We can move on to conduct the experiment to find out the optimal scenario.

1. Experiment

We set an experiment. The run length is 8 days. Warm-up period is 1 day. Max replications is 10. Min replications is 6. Max scenarios is 25. We add two responses and name them TC and AverageWaitingTime. Set the expression for TC as totalcost.Value. The objective is Minimize. Set the expression for AverageWaitingTime as avgwaitingtime.Valye and objective as Minimize. The property is shown below.

Graphical user interface, table

Description automatically generatedTable

Description automatically generated

Table

Description automatically generated

Set each of the control Minimize Value as 1 and Maximize Value as 5 and increment as 1.

Below is an example of the property setting and keeps the same for other controls.

Table

Description automatically generated

Then we found out all our scenario doesn’t work because some of the servers’ utilization is achieve 100%, which means the system cannot handle the current passengers. So, we decide to run another experiment. Also, we notice that the servers that has gotten crushed are Dropoff, BaggageCheck, BaggagePickup, WTMD. Probably because all of those servers have to be went through by all the passengers, while other servers will only go through part of passengers.

So, in the experiment 2, the run length is 8 days. Warm-up period is 1 day. Max replications is 10. Min replications is 6. Max scenarios is 20. We add two responses and name them TC and AverageWaitingTime. Set the expression for TC as totalcost.Value. The objective is Minimize. Set the expression for AverageWaitingTime as avgwaitingtime.Valye and objective as Minimize. The property is shown below

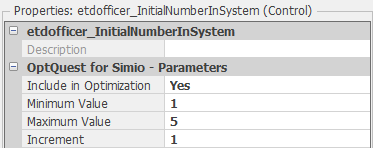
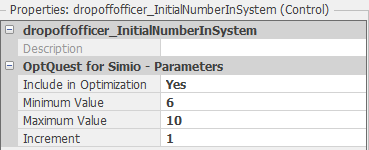
Table

Description automatically generated

Then for dropoffofficer\_InitialNumberInSystem, WTMD\_InitialCapacity, BaggagePickup\_InitialCapacity, BaggageCheck\_InitialCapacity, we set the minimum as 6, maximum as 10 and increment as 1.

For etdofficer\_InitialNumberInSystem, patdownofficer\_InitialNumberInSystem, WTMD2\_InitialCapacity, WTMDCheck\_InitialCapacity, we set the minimum as 1, maximum as 5 and increment as 1.

Below is the example of setup.



1. Simulation result Analysis

In experiment 1, when we set the minimum value for all controls as 1 and maximum value for all controls as 5, we found out that there are always some servers crushed in all scenarios. So, that means the capacity of certain servers is not enough. Also, because all scenarios cannot handle the system, there is no way we can find the optimal among those scenarios. So, we did experiment2. The detail of experiment setup is as above in part 3.

Since we have two responses, total cost and average waiting time. We would like to balance the relationship between total cost and total waiting time. So, we gathered all the total cost and total waiting time from the scenarios that did not crush and put the data into a data as shown below. If a scenario’s utilization achieves 100, we put “Yes” in the first column, otherwise put “No” in the second column. Then the following two columns are total cost and total waiting time.

Then final cost is calculated by 4\*total waiting time + total cost. The function means if the passengers wait more than 1 minute, then it would cost $4 more.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | (Utilization>=100)YES | No | Total Cost | Total Waiting Time | Final Cost |
| Scenario1 |  | no | 35678.8 | 178.90623 | 36394.42 |
| Scenario2 | yes |  |  |  | 0 |
| Scenario3 |  | no | 50872.3 | 203.48955 | 51686.26 |
| Scenario4 | yes |  |  |  | 0 |
| Scenario5 |  | no | 43275.6 | 192.27767 | 44044.71 |
| Scenario6 |  | no | 23315.3 | 251.16374 | 24319.95 |
| Scenario7 | yes |  |  |  | 0 |
| Scenario8 | yes |  |  |  | 0 |
| Scenario9 |  | no | 38294.6 | 210.5989 | 39137 |
| Scenario10 |  | no | 33140.8 | 173.89008 | 33836.36 |
| Scenario11 | yes |  |  |  | 0 |
| Scenario12 |  | no | 23296.3 | 248.3561 | 24289.72 |
| Scenario13 | yes |  |  |  | 0 |
| Scenario14 |  | no | 20815.3 | 250.9063 | 21818.93 |
| Scenario15 |  | no | 23296.3 | 253.4901 | 24310.26 |
| Scenario16 | yes |  |  |  | 0 |
| Scenario17 |  | no | 23159.8 | 223.3206 | 24053.08 |
| Scenario18 | yes |  |  |  | 0 |
| Scenario19 | yes |  |  |  | 0 |
| Scenario20 | yes |  |  |  | 0 |
| Scenario21 |  | no | 20621.8 | 220.8666 | 21505.27 |
| Scenario22 |  | no | 20621.8 | 220.8666 | 21505.27 |
| Scenario23 | yes |  |  |  | 0 |

As I indicated in the table by yellow, scenario 21 and 22 are the optimal solutions for the airport. The total cost for wage and equipment fee is $20621.8 and the total waiting time in 8 days is 220.866 minutes. They are the same scenario except that they have different WTMD\_CheckInitialCapacity. But the number of patdown officer is the same. The specific number of officers and equipment are as following.

Dropoffofficer\_InitialNumberInSystem : 6

Etdofficer\_InitialNumberInSystem: 1

Patdownofficer\_InitialNumberInSystem: 1

WTMD\_InitialCapacity: 7

WTMD2\_InitialCapacity: 1

BaggageCheck\_InitialCapacity: 8

BaggagePickup\_InitialCapacity: 7

This is the optimal choice for the airport to minimize final cost.

1. Recommendations, Benefits, and obstacles

By modeling the baggage drop-off, baggage pick-up, WTMD, ETD etc., we have analyzed various scenarios to support decision making processes in the passenger security checkpoint by studying the effects of changes in variables, such as how many officers to keep when multiple X-ray machines are operating in the same time and what type of queuing system should be applied. The improvements can be made regarding the behavior of the check-in system. This may prove to be a helpful tool in airport management systems or an airline to forecast, optimize and make important efficiency in passenger level of service.

The advantages of improving simulation-based airport security checkpoints model include the ability to provide immediate feedback, repetitive practice learning, the ability to adjust the difficulty level, and the adaptability to different types of learning strategies. By hiring more ETD faculties or pat down faculties, the airport system can be modeled to study the effects of efficiency parameters such as number of passengers on a flight and check-in counter opening and closing time, etc.

Whatever considering adding the extra metal detector or more pat down and ETD officers to decrease the queue up or adding extra officers or X-ray machines for the baggage process to reduce the time, the airport needs to spend more money on the maintaining fees or salaries on their budget. Especially during the COVID-19 period, the airport faculties are the high-risk infectious people while the airport also meets the budget crunch.