IE324 Simulation
2022 Spring Term Project
Round 2

Simge Çınar, 21901465

Sena Ceren Göktaş, 21704202

Yağız Yaman, 21702889

Logic of the Model

Creating Call Entities:

Firstly, we started with three separate create blocks which are named "Create Calls From Ferry", "Create Calls From Train" and "Create Calls From Other." The time between arrivals is determined by schedule in each of the Create blocks. We have used the true arrival rates and distributions that are shared after the submission of the first round. The inter arrival rates of the Poisson distribution are defined with the rate of the given exponential distributions. Each rate in four hourly period calculated as $\frac{60}{\lambda}$ for an hour.

Usage of Assign Blocks at Start:

I. **Location_x & Location_y**
We have assigned these attributes to each entity to determine the location they have called from the grid map which is 14x17. Some of the locations are pre-determined, such as the locations of Ferry Station (14,17) and Train Station (14,1). In those cases, we just used these attribute values. However, for the Other Locations, we needed to randomly assign those locations. Therefore, we used the expressions *ANINT(UNIF(1,14))* and *ANINT(UNIF(1,17)).* These expressions assigned a value to the locations between the defined intervals. Yet, since UNIF() is a continuous random distribution, we used ANINT function to round it to the nearest integer.

II. **Destination_x & Destination_y**
We assigned the location of the destinations by using the same logic. Additionally, we used a decide block to assign destinations for the calls from the Other Locations on the map.

III. **Group Size**
We used the DISC() function to assign the group size attribute to each call entity since group size's distribution is discrete.

Decide Block on Destination and Location:

After the assignment of the attributes to the entities, we needed to check if the randomly assigned destination attribute values are the same with the randomly assigned location values. We used a decide block to check whether it is the case. These blocks basically check whether the x and y pairs of the map are the same or not by expression *"Location_x == Destination_x && Location_y == Destination_y"*. If they are the same, it returns to the Assign blocks and new values assigned. Since there is no delay between the assign and decide blocks, the TNOW values will not be affected by this new assignment.

Number of Cars Needed:

We assigned the number of the cars needed using a decide block in order to seize the correct number of the cars that the customer needs from the dispatch centers. This block checks the group size of the entities and according to the group size it connects to the Assign blocks. If the group size is smaller than or equal to 4, the Cars Needed attribute takes the value of 1. If it is between 4 and 8, Cars Needed takes the value of 2. Otherwise, which has only a one group size possibility left, Cars Needed takes the value of 3.

Determination of the Nearest Center:

There are some specific order of conditions to seize the required number of cars from the dispatch centers. In order to check those conditions, firstly we needed to determine the closest dispatch center to the current locations of the customers. We used a decide block which compares the distances (found by taking absolute value of the difference of the points on map, multiplied by the corresponding length which is 3 for the vertical and 1 for the horizontal lines) and takes the smallest one as the nearest center.

Which Dispatch Center to Send from?

After the determination of the nearest dispatch center, the same logic given in the question followed. Three Decide blocks have been used to emphasize the priority of the conditions. The first Decide block checks whether there are enough number of cars in the nearest center by *"40 – NR(DC1 Vehicle) >= Cars Needed"*. It checks how much the capacity of the vehicles in one center, which is 40, is used currently and if it is greater than the assigned car number. If yes, let us assume that the nearest is Dispatch Center 1 (DC1), then it goes to DC1. If there are not enough cars, then it checks the same condition for the second dispatch center which is the far one through the Else exit of the block. Lastly, if there is no available car in the second center as well, then another Decide block checks whether all the cars are busy or not. If yes, then the call is canceled with probability 70%. The calls that are not canceled, which are willing to wait, then matched with the nearest center it has. This whole procedure is also repeated for the other dispatch center as well.

I. **Canceled? 1 & Canceled? 2**
   In the case where there are no idle cars at both of the centers, the calls are canceled with determined probabilities. These decide blocks are used to return the calls that are not canceled to the dispatch centers and the ones that are canceled to the dispose block. These entities are assigned as "Unsatisfied" entity type since they are lost customers and it will be needed in the statistics of the model. Then, it is counted with record blocks and disposed through the system.

Usage of Assign Blocks after Deciding the Dispatch Center:

We used three Assign blocks after deciding from which center the vehicle will be sent. We used them to determine the departure time from the dispatch center, the time the client was picked up and to determine the waiting time accordingly.

I. **Assign DC1 & Assign DC2:**

   If the vehicle is going to be sent from dispatch center 1, we assign an attribute called Selected DC in the Assign block and set its value to 1. If the vehicle is going to be sent from dispatch center 2, we set the value of pick up in Assign block to 2.

II. **Time Before & Time Before Cont:**

   We determined the departure time from the dispatch center to pick up the customer using TNOW function and named it as TimeBefore and TimeBeforeCont for the dispatch center 1 and dispatch center 2 respectively.

III. **Waiting Time & Waiting Time Cont:**

   We find the waiting time by subtracting the current simulation time from the departure time. The expression was written as *"TNOW-TimeBefore"* and named Waiting time for dispatch center 1. For dispatch center 2, the expression was written as "TNOW-TimeBeforeCont" and named WaitingTimeCont.

Seize Blocks for Resource Vehicles:

In order to send the vehicles from the Dispatch Centers (DC), we need to seize it first. We achieved it by putting a Seize block. The number of vehicles to be seized is determined by the previous steps of the model and saved as an attribute called Cars Needed. We used this amount to seize the resources "DC1 Vehicle" and "DC2 Vehicle" with each having a capacity of 40. We used two seperate seize blocks to create two different queues.

Delay Blocks for Arrival to Location:

We created some expressions through the expression module to determine the delay time for each entity. We calculated the delay time of the expressions "Arrival Time DC1" and "Arrival Time DC2" by calculating the expression value as *(3*ABS(13 - Location_y) + ABS(7 - Location_x)) / Speed*. This finds the absolute differences between the DC1 (for the DC2 the points 13 and 7 changes respectively) and the randomly assigned location point. After finding the distance, it calculates the distance in block length by multiplying the block distances with 3 for the y-axis and 1 for the x-axis as it is given in the question. The speed of the vehicles changes due to traffic density and its unit is block/min. Therefore, we created a variable called "Speed" and used it in the expression formula as well in the denominator. The delay times' units are also in minutes.

Dummy Logic for Speed:

We created a variable called Speed and set its initial value to 6.5. Since the value of speed changes every 4 hours during the day, we used a dummy logic to enable its change. We added a Create block apart from the model, chose the type of the time between arrivals as constant and set its value to 240 (4 hours in minutes) to create only one entity in every 4 hours. Additionally, we set the first creation time as 240 as well since the initial value will change firstly in the 4th hour.

Then, we used an Assign block to determine which time interval the current simulation time corresponds to among the 6 equally divided intervals of the speed (hours 1-4, 5-8, and so on.), which assigns an attribute called "Speed Interval". The logic behind this is to be able to assign correct speed values according to these intervals through a Decide block. Hence, we used the expression value *ANINT(MOD((TNOW/240),6))*. This firstly divides the current simulation time (TNOW) to 240 (since our base time is in minutes, we used the value of 240) and finds the value in terms of 4 hourly units. Then, it takes the mod 6 of this value (since there are 6 time intervals) and rounds it to the nearest integer using ANINT function.

After finding the correct interval and its value, the Decide block checks the "Speed Interval" values and connects to the corresponding Assign block which will change the value of the Speed variable according to the interval determined. To exemplify, if *"Speed Interval == 1"* then the decide block connects to the "Assign Speed 2" block then changes the value of the Speed to 4.5 block/min. After these changes, each entity leaves the system through the "Dispose Dummy Entity" block.

Determining Satisfaction:

The determination of the satisfaction depends on the arrival time of the vehicle to the customer's calling time. If it is less than 10 minutes, the customer is considered as satisfied. If it is between 10 and 15 minutes, then the customer is partially satisfied. However, if it is more than 15 minutes, the customer is considered a lost customer. We used an Assign block called "Time Before" and "Waiting Time" to determine these. The "Time Before" block is put before the Seize block since we wanted to add the wait time in the queue (if there is any) to the waiting time of the customer. "Time Before" block assigns the TNOW as TimeBefore and "Waiting Time" assigns the WaitingTime attribute according to the value of *TNOW-TimeBefore* after the queue time and delay time of the arrival. Then, using Decide blocks, we check the Waiting Time according to the defined minute intervals. Hence, We changed the entity type of the call entities by "Assign Unsat", "Assign Partially Sat" and "Assign Sat".

Recording Satisfaction:

We counted those entities via Record blocks to see the statistics regarding the customer satisfaction levels. In addition to these counts, the satisfactory level is also determined as “Unsatisfied” on the side where the customer cancels the ride if all the vehicles are busy and then counted through the same “Record Unsat”.

Arrival to the Destination:

The satisfied and partially satisfied customers picked up from their locations and then for the arrival time of the customers to their desired destination is calculated through the same expression module logic. The “Arrival to Destination” Delay block is used to show this delay. The value of the “Arrival Time Dest” expression calculated by *(3*ABS(Location_y- Destination_y) + ABS(Location_x - Destination_x)) / Speed.* It is using the randomly chosen location and destination attributes to calculate the distance as before. However, it should be noted that the entities defined as “Unsatisfied” due to canceled calls or arrival time exceeding 15 minutes are not passing through this Delay block. The vehicles do not leave the center due to the canceled calls. Additionally, we have assumed that the vehicle goes to the location and sees the customer left when the arrival time is greater than 15 minutes. Therefore, there is no destination delay in both cases. Additionally, it should be noted that after the customers dropped to their destinations, the entities continued to represent the vehicles.

Decide Blocks for Where to return?

A Decide block is used for the cases where the customer is carried to their destination and for the vehicle sees no customer is waiting on the location where the call happened. In both cases, the vehicles should return to the same DC that they have been sent from. We used the same attribute called “Selected DC” that we have determined before seizing the resources. If *“Selected == 1”*, it is returned to DC1 and if *“Selected == 2”*, it is returned to DC2.

Delay Blocks for Return to DC’s:

The same expression module logic has been used in the determination of the delay in the Delay modules called “Return to DC1 Uns”, “Return to DC2 Uns”, “Return to DC1” and “Return to DC2”. It uses the difference between the location and the dispatch centers for the unsatisfied customers such as *(ABS(Location_x - 7) + 3 * ABS(Location_y - 13)) / Speed* (for the DC1, which is stored as “Return Time to DC1 Uns”) and it uses the difference between the destination and the dispatch centers for the satisfied and partially satisfied customers such as *(ABS(Destination_x - 7) + 3 * ABS(Destination_y - 13)) / Speed* (for the DC1 again, which is stored as “Return Time to DC1”). The same procedure also followed for the DC2.

Release Blocks and Disposal of Entities:

After the delay, which represents the time that the vehicles turn to their dispatch centers, a Release block is put to release the resource vehicles. This block releases the number of cars that are sent for the customer. Lastly, the Disposal blocks are used for dispose of call entities.

Extra Assumptions

- If the nearest center is DC1 and there’s no idle vehicle in both centers, the customers wait for a vehicle from DC1. The same applies for DC2.
- We assumed that a taxi can only earn money while taking a customer to his/her desired destination. A taxi cannot earn money while it goes to pick up the customer from where he/she is or while going back to the dispatch center after leaving the customer. Hence, we did not include all the distances the taxis made. So, we only included the distance that was made while taking a customer to his desired destination.

Statistics

Profit: -$512912

Total Distance Traveled: 2010471

Total number of satisfied customers: 21755

Total number of partially satisfied customers: 688

Total number of unsatisfied customers: 3787

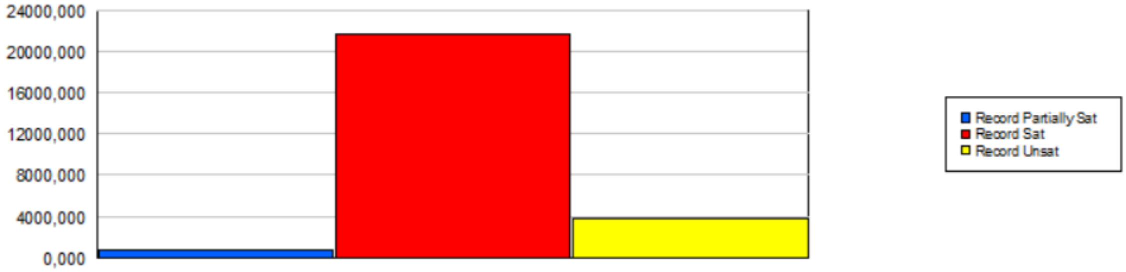
Average time a vehicle spends from dispatch center to pick up point: 12.4256

Average time a vehicle spends from pick up point to destination point: 6.8695

Average time a vehicle spends from destination point to dispatch center: 10.5683

Counter

Count	Value
Record Partially Sat	688.00
Record Sat	21755.00
Record Unsat	3787.00



Time Persistent

Time Persistent	Average	Half Width	Minimum Value	Maximum Value
DC to pick up	12.4256	(Correlated)	0.9231	39.2000
Destination to DC	10.5683	0,915868231	4.6154	27.2000
Pick up to destination	6.8695	0,622088818	0.00	24.4000

Output

	Value
Profit	-512912.00
Total Distance Travelled	2010471.00