Sharif University of Technology

Industrial Engineering Department

Simulation and Performance Evaluation of a Car Insurance Assessment Center

Sadra Daneshvar[1] 98104111

Soroush Etminanbaksh[2] 98103891

Course: **Princles of Simulation**

Professor: **Dr. Nafiseh Sedghi**[3]

April, 2023

[1] Sharif University of Technology, Tehran, Iran: amirsadra.daneshvar@ie.sharif.edu

[2] Sharif University of Technology, Tehran, Iran: soroush.etminan@ie.sharif.edu

[3] Sharif University of Technology, Tehran, Iran: nafiseh.sedghi@sharif.edu

Abstract

Simulation is a method of approximation that involves modeling an operation, process, or system and characterizing its behavior over time. This technique is commonly employed for the purpose of analyzing, measuring, and predicting changes and their subsequent impact on important performance metrics such as cost, quality, and time. By utilizing simulation techniques, researchers and practitioners can gain valuable insights into the underlying dynamics of complex systems, and make informed decisions to improve their performance.

This project adopts the discrete-event systems simulation approach to model a car insurance assessment center that involves multiple activities carried out by various experts. The simulation model will utilize statistical models and performance metrics, such as maximum queue length and average waiting time, to assess the center's performance and identify opportunities for optimization. The objective of this project is to enhance the operational efficiency of the car insurance evaluation center through simulation-based insights and recommendations.

Keywords: simulation, discrete-event simulation, statistical models, queue length, waiting time, car insurance

Table of Contents

Table of Figures

Chapter One; Phase 1

1 Problem Statement

This study presents the problem of a car insurance assessment center that operates from 8:00 am to 6:00 pm, and has a limited capacity for processing cars. If a car arrives after the closing time, it is not serviced until the next working day. Cars arriving without a pair must wait until their partner arrives, and are given priority over single cars. The queue for taking photos has a limited capacity, and cars may have to wait outside until space is available. The filing process involves three experts, each with a triangular service distribution, while the complaint registration process has an exponential distribution with an average duration of 15 minutes. Completed cases are given priority over those still in progress, and cars need to repeat the filing process after registering a complaint.

Numerous performance assessment parameters can be envisaged for this system, such as the mean queue length or experts' idle time. The present study utilizes discrete-event simulation to model and obtain these statistical measures. Moreover, the implementation model and its outcomes will be scrutinized using the Python programming language.

1.1 System Modeling and Static Description

Before simulating any problem, a static description of the system being investigated must be provided, including the identification and definition of state variables, entities, events, and overall system characteristics. This description serves only to recognize the system, without describing its performance or progress. In the following section, the modeling assumptions are presented followed by the static description of the system.

1.1.1 State Variables

State variables are fundamental components that define the state of a system at any given moment and ultimately assist in gathering crucial statistics and results. Within the context of this modeling, a comprehensive set of state variables has been identified, including:

1. $Q_{Out}\ (t)$: Outside's queue length at time t
2. $Q_{Ph}\ (t)$: Photography's queue length at time t
3. $Q_{Pa}\ (t)$: Parking's queue length at time t
4. $Q_{Doc}\ (t)$: Documentation's queue length at time t
5. $Q_{Eval}\ (t)$: Evaluations's queue length at time t
6. $Q_{Com}\ (t)$: Complaint's queue length at time t
7. $Q_{Ful}\ (t)$: Fulfillment's queue length at time t
8. $P_i\ (t)$: i^{th} photographer's status at time t (0: idle, 1: busy)
9. $S_{DF_i}\ (t)$: i^{th} documneter/fulfiller expert's status at time t (0: idle, 1: busy)
10. $S_C\ (t)$: Complaint expert's status at time t (0: idle, 1: busy)
11. $S_{E_i}\ (t)$: i^{th} evaluation expert's status at time t (0: idle, 1: busy)

1.1.2 Entities

As the servers in each section are similar, it is unnecessary to label and create a separate entity for them. However, in the case of clients (defined as each pair of vehicles), it is imperative to identify and assign certain attributes. Vehicle's entrance number (i) needs to be determined along with whether they enter alone or as a group (r $[0: alone, 1: paired]$). Additionally, it is essential to establish whether they will file a complaint or not (c $[0: doesn't\ file, 1: files]$). Based on this information, the following entity can be established: $\boldsymbol{V_{irc}}$

For instance, a V_{710} shows the 7th vehicle that has arrived with a partner and does not want to file a complaint.

1.1.3 Events

In this modeling, events play a crucial role in altering the state of the system by modifying one or more of its state variables. Hence, it is essential to identify and classify the events that occur in the system. The following events have been considered in this study:

1. Shift Start (SS): It indicates the beginning of the work shift at the beginning of each day
2. Arrival (A): This signifies that the vehicle has arrived at the assessment center
3. Entry (E): It indicates the entry of vehicles into the premises of this center
4. Partner Arrival (A_P): It indicates the arrival of the partner of a car that has come alone beforhand
5. Photography Departure (D_P): It indicates the end of the photography process and the departure from that section
6. Documentation Arrival (A_D): It indicates arrival at the documentation section
7. Documentation Departure (D_D): It indicates the end of the documentation process and the departure from that section
8. Evaluation Arrival (A_E): It indicates arrival at the evaluation section
9. Evaluation Departure (D_E): It indicates the end of the evaluation process and the departure from that section
10. Complaining Arrival (A_C): It indicates arrival at the complaning section
11. Complaining Departure (D_C): It indicates the end of the complaning process and the departure from that section
12. Fulfillment Arrival (A_F): It indicates arrival at the fulfillment section
13. Fulfillment Departure (D_F): It indicates the end of the fulfillment process and the departure from that section
14. Shift End (SE): It indicates the end of the work shift

It is noteworthy that in section 0, a comprehensive explanation of all events and their respective logic is presented, accompanied by flowcharts that depict their flow.

1.1.4 Event Notices

In this section, we state the event notices for each event that will be used in the Future Event List (FEL):

1. (SS, t): Shift starts at t
2. (A, V_{irc}, t): ith pair of vehicles arrive at t
3. (E, V_{irc}, t): ith pair of vehicles enter at t
4. (A_P, V_{irc}, t): the partner of ith pair of vehicles arrives at t

5. (D_P, V_{irc}, t): i[th] pair of vehicles depart from photography section at t
6. (A_D, V_{irc}, t): i[th] pair of vehicles arrive at the documnetation section at t
7. (D_D, V_{irc}, t): i[th] pair of vehicles depart from documentation section at t
8. (A_E, V_{irc}, t): i[th] pair of vehicles arrive at the evaluation section at t
9. (D_E, V_{irc}, t): i[th] pair of vehicles depart from evaluation section at t
10. (A_C, V_{irc}, t): i[th] pair of vehicles arrive at the complaining section at t
11. (D_C, V_{irc}, t): i[th] pair of vehicles depart from complaining section at t
12. (A_F, V_{irc}, t): i[th] pair of vehicles arrive at the fulfillment section at t
13. (D_F, V_{irc}, t): i[th] pair of vehicles depart from fulfillment section at t
14. (SE, V_{irc}, t): It indicates the end of the work shift

1.1.5 Lists

Lists play a critical role in simulations as they allow for efficient and organized storage of data. In a simulation, lists can be used to store various entities such as customers, servers, and events. The following lists are essential for conducting this simulation:

1. $L_{Q_{Out}}$: A compiled list of vehicles which are currently waiting in outside's queue, arranged in chronological order based on their respective arrival times.
2. $L_{Q_{Ph}}$: A compiled list of vehicles which are currently waiting in photographer's queue, arranged in chronological order based on their respective arrival times.
3. $L_{Q_{Pa}}$: A compiled list of vehicles which are currently waiting in parking's queue, arranged in chronological order based on their respective arrival times.
4. $L_{Q_{Doc}}$: A compiled list of vehicles which are currently waiting in documnetation's queue, arranged in chronological order based on their respective arrival times.
5. $L_{Q_{Eval}}$: A compiled list of vehicles which are currently waiting in evaluation's queue, arranged in chronological order based on their respective arrival times.
6. $L_{Q_{Com}}$: A compiled list of vehicles which are currently waiting in Complaint's queue, arranged in chronological order based on their respective arrival times.
7. $L_{Q_{Ful}}$: A compiled list of vehicles which are currently waiting in fulfillment's queue, arranged in chronological order based on their respective arrival times.

1.1.6 Activities

Activities play a crucial role in a simulation as they represent the actions that entities take within the system being modeled. By defining activities, the simulation modeler can control the flow of the simulation and accurately represent the real-world system. The activities required for this simulation are as follows:

1. Interarrival Time: Denoted by a_{ij} where $i \in (1\!: Rainy, 2\!: Snowy)$ and $j \in (1\!: First\ Period, 2\!: Second\ Period, 3\!: Third\ Period, 4\!: Fourth\ Period)$
2. Waitning Time: Denoted by w for how long is the partner going to take before arriving
3. Photography Service: Denoted by s_p for each photographer
4. Documentation/Fulfillment Service: Denoted by s_{df} for each expert
5. Complaining Service: Denoted by s_c for each expert
6. Evaluation Service: Denoted by s_e for each photographer

1.1.7 Delays

Delays play a crucial role in simulations as they help to model realistic scenarios where systems do not function at full capacity. Accounting for delays in a simulation helps to identify potential bottlenecks in a system and evaluate the impact of these bottlenecks on overall system performance. In this simulation, the delay refers to the duration of time that each pair of cars spends in the system while waiting in queues.

1.1.8 Initial State

At the onset of the simulation, the state variables are initialized to their respective initial states, as elaborated below. It is further assumed that the first vehicle enters the system at $t = 0$. Consequently, the future event list (FEL) at the commencement of the simulation is as follows: $FEL: \left((A, V_{100}, 0)\right)$

1. $Q_{Out}\,(0) = Q_{Ph}\,(0) = Q_{Pa}\,(0) = Q_{Doc}\,(0) = Q_{Eval}\,(0) = Q_{Com}\,(0) = Q_{Ful}\,(0) = 0$
2. $P_i\,(0) = S_{DF_i}\,(0) = S_C\,(0) = S_{E_i}\,(0) = 0$
3. $L_{Q_{Out}} L_{Q_{Ph}} = L_{Q_{Pa}} = L_{Q_{Doc}} = L_{Q_{Eval}} = L_{Q_{Com}} = L_{Q_{Ful}} = \phi$

1.1.9 Premises and Justifications

Fortunately, the problem under investigation did not involve any significant ambiguity that necessitated significant changes or assumptions. However, it is important to note the following points regarding the system's modeling:

- Given that the documentation and fulfillment experts are the same, a state variable is designated for them, and they are not treated separately.
- The events are named with "Arrival" and "Departure" since these terms accurately describe the nature of the events and provide clear terminology for the reader.
- The "Fulfillment Departure" event marks the end of each pair of vehicle’s route and serves as the general departure event, which triggers the vehicle's departure from the center.
- At the simulation's start, the paired/unpaired and complaint/non-complaint indices for each input entity are in state zero ($r = 0 \; and \; c = 0$). These indices will be updated during the simulation if there are any changes.

1.2 Dynamic Description and Flow Charts

The dynamic description of a system involves not only the recognition of the system but also the examination of its progress over time, or other factors depending on the modeling approach. This process is considered continuous, but due to the discrete nature of the events, the progress occurs at certain points. In this section, we will explore the dynamic description of the system in question, and provide a flow diagram for each of the defined events. This will provide a better understanding of the system's behavior and performance under different scenarios.

1.2.1 Shif Start (SS)

The initial event in the simulation marks the start of the work shift. At t=0, the weather condition for the day is determined, the length of all queues is reset to zero, and the lists and statistics are cleared. This routine is crucial to maintain the accuracy and reliability of the simulation. Undertaking these measures at the beginning of each shift ensures that the simulation functions appropriately and provides valid outcomes.

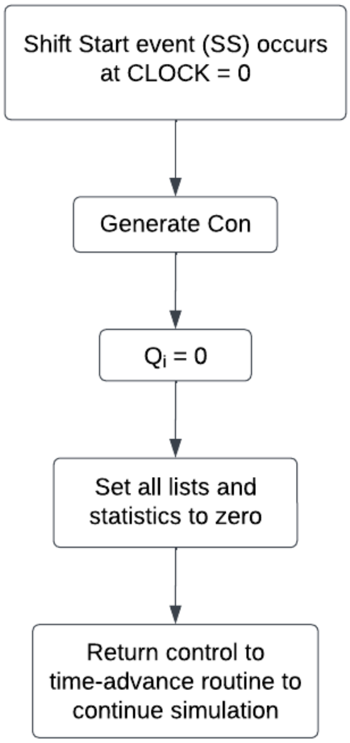

Figure 1- Shift Start Event Flowchart

1.2.2 Arrival (A)

In this event, upon the arrival of each car, the system first determines whether it is accompanied by a partner or not. If it is a pair ($r = 1$), the availability of the shooting queue is checked. If the queue has available space, the entry event is scheduled for the same time (t). However, if there is no space, the length of the outside area queue is increased. If the car arrives alone, the system generates the arrival time of its partner based on the given distribution, and schedules the partner arrival event at the time of the companion's arrival. Then, if there is room in the photo queue, the single car proceeds to the parking lot and is added to the queue and list. Otherwise, it is added to the queue and list. Finally, the system checks the weather conditions and generates the appropriate interarrival time based on the time of day for the next pair of cars to arrive, thus enabling the system to plan ahead.

1.2.3 Entry (E)

At the onset of this event, the presence of members in the parking queue is verified. If a member exists, they are removed from the queue and it is then determined whether this pair of cars can commence shooting after entering the area. Initially, the status of the first specialist is assessed. If they are idle, their status is updated to busy, a service novel is created, and the Photography Departure event is scheduled. If the first specialist is busy, the same process is carried out for the second expert. If both specialists are occupied, the pair of cars is added to the photo-taking queue.

In the event that the parking lot is empty, the first member of the queue outside is extracted. Following this, the incoming machine's pairing status is checked. If it is paired, the identical process described earlier is executed. However, if it is not paired, the machine is appended to the queue and the parking list.

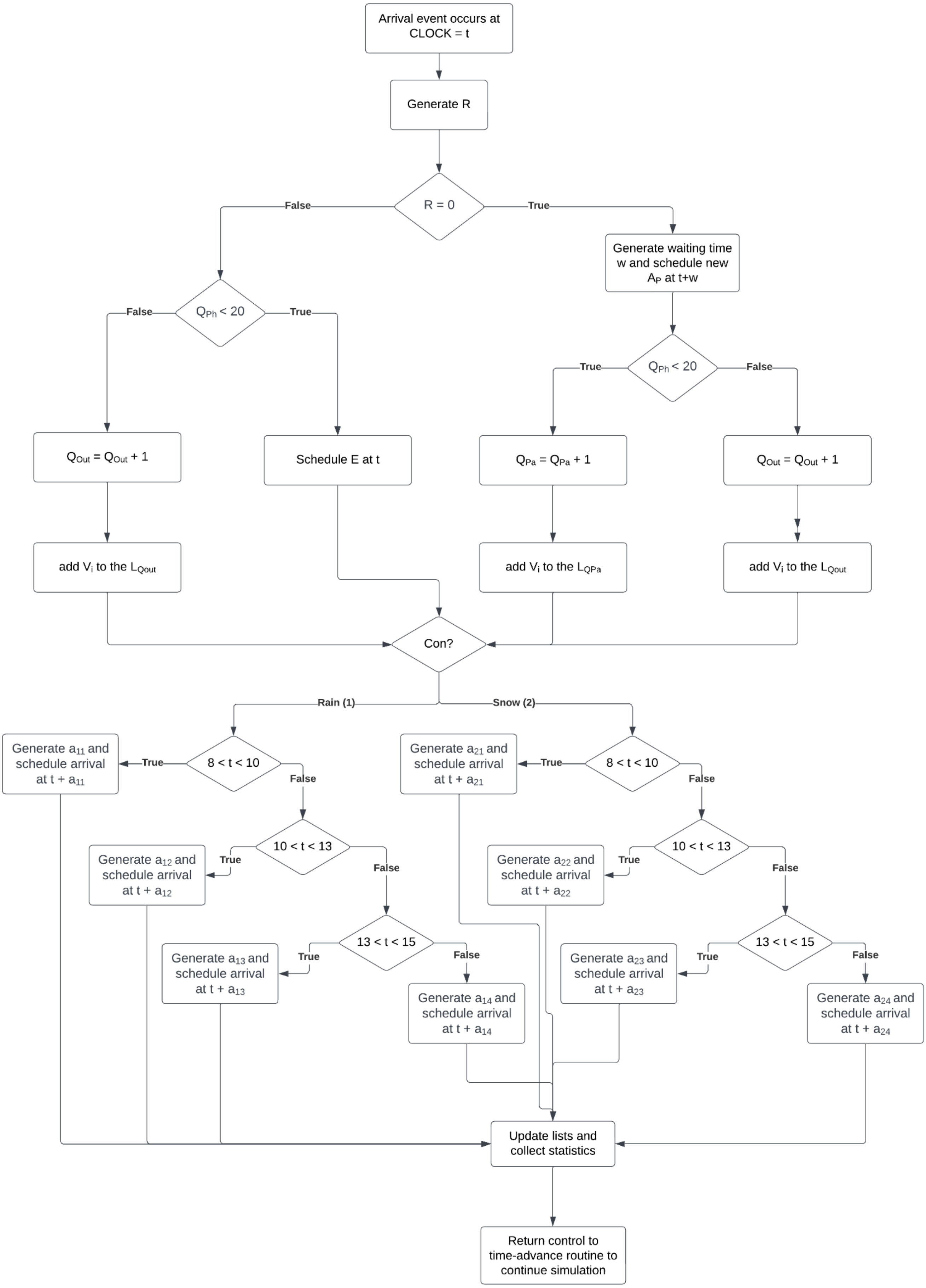

Figure 2- Arrival Event Flowchart

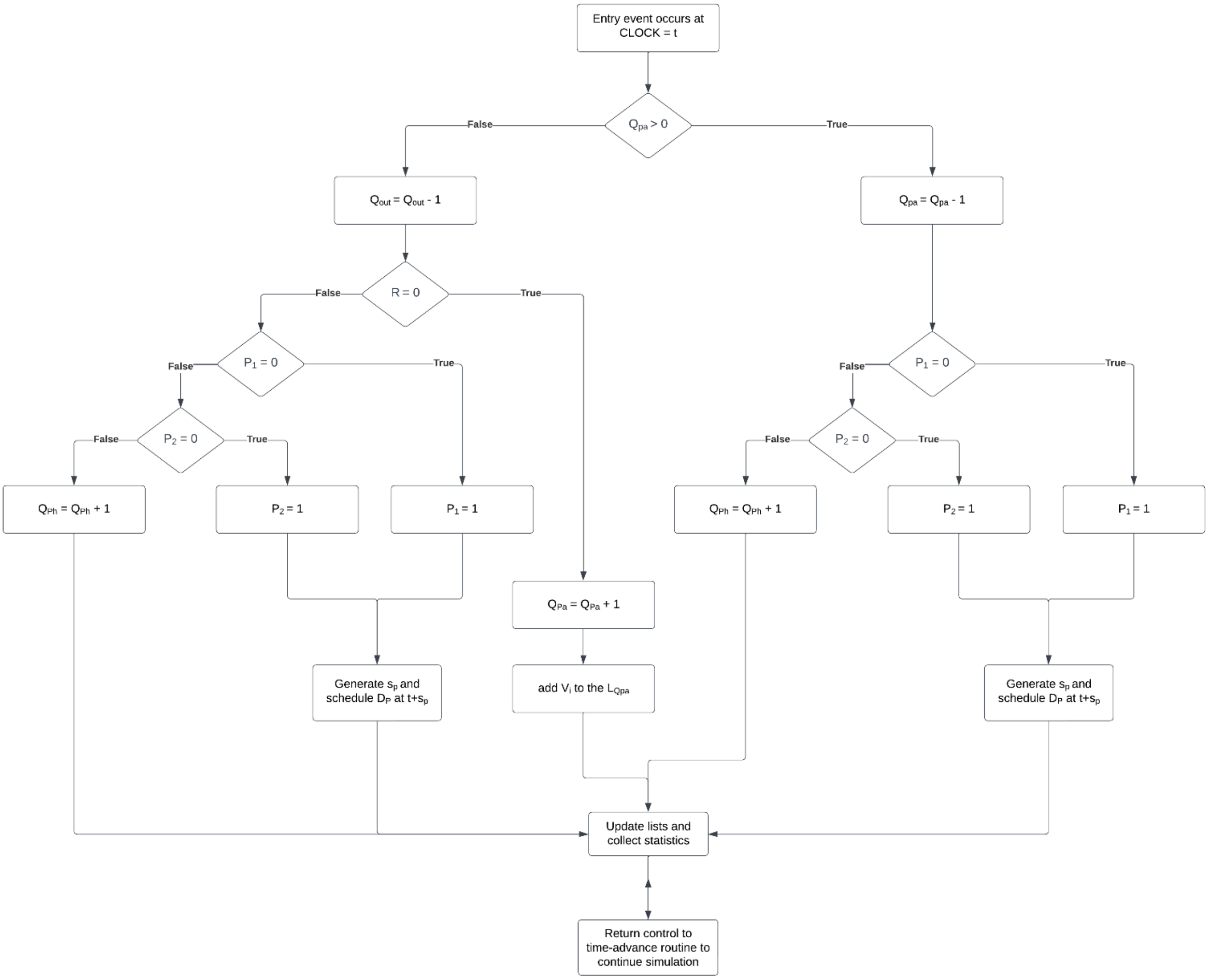

Figure 3- Entry Event Flowchart

1.2.4 Partner Arrival (A_P)

At the onset of this event, a check is conducted to determine the location of the arriving vehicle, whether it is inside the parking lot or not. If it is established that the car is inside the parking lot, an entry event will be scheduled concurrently. However, if the vehicle is outside the parking lot, it is inferred that the car is still in the queue, and hence, it is sufficient to alter its state and assign it a pair state ($r = 1$).

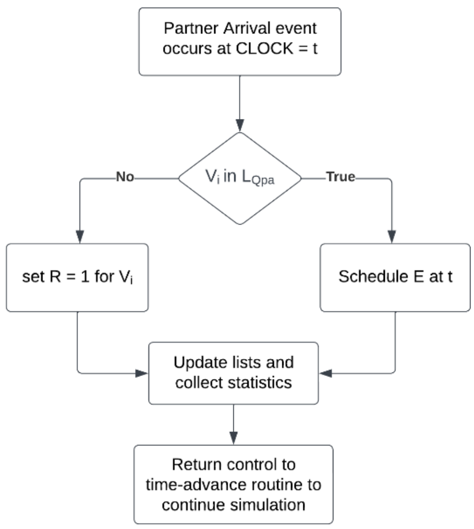

Figure 4- Partner Arrival Event Flowchart

1.2.5 Photography Departure (D_P)

In this phase of the problem, a preliminary check is conducted to ascertain the existence of a photography queue. Should there be a queue, the queue length is decremented, the service time is computed, and a scheduled photography departure event is initiated. However, if the queue is absent, the server status is changed to idle. Upon completion of these processes, an Arrival Documentation event is scheduled.

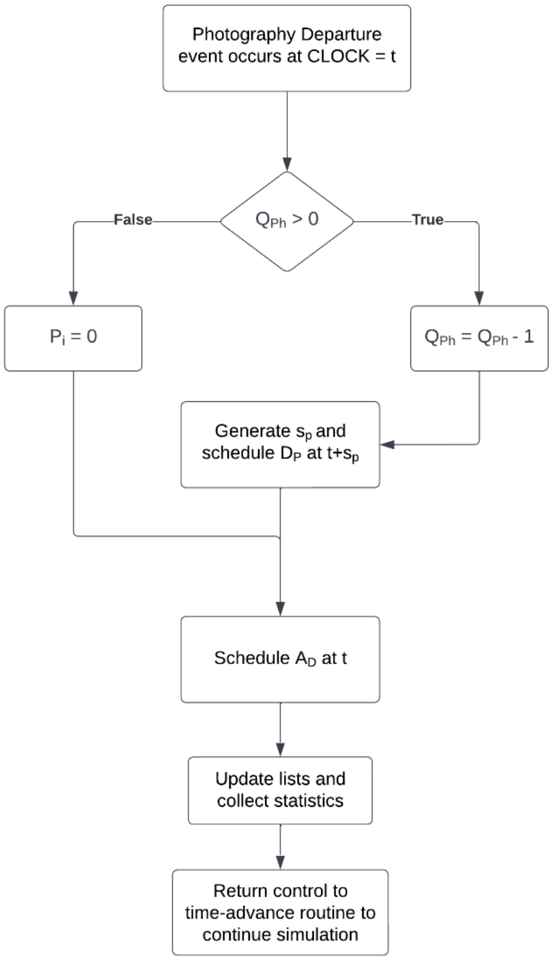

Figure 5- Photography Event Departure Flowchart

1.2.6 Documentation Arrival (A_D)

In this particular event, the system first examines the availability of the queue of cars exiting the Fulfillment department, given that they have a higher priority. If the queue is full, then a car is added to the Documentation queue. On the other hand, the system checks the availability of the first specialist. If the specialist is idle, they are assigned a service time and the Documentation Departure event is scheduled. However, if the specialist is occupied, the same process is conducted for the next specialist in line. This process is repeated until an available specialist is found. If all three specialists are occupied, the pair of cars is added to the Documentation queue.

1.2.7 Documentation Departure (D_D)

In this particular phase of the problem, the queue status of Documentation is initially verified to determine if any customers are awaiting service. If the queue is occupied, one customer is removed from it, the corresponding service time is evaluated, and a Documentation Departure event is scheduled accordingly. Conversely, if no customers are queued, the server status is updated as "idle." Following these procedures, an Evaluation Arrival event is scheduled to occur.

1.2.8 Evaluation Arrival (A_E)

In this section of the problem, the status of the first specialist is checked for availability. If the specialist is found to be unoccupied, their status is changed to "busy" and a service time is assigned to them. The Evaluation Departure event is then scheduled for the specialist. If the first specialist is occupied, the same process is repeated for the second specialist. If both specialists are occupied, the pair of cars is added to the Evaluation queue for further processing.

1.2.9 Evaluation Departure (D_E)

In this segment of the problem, the first step entails checking the status of Evaluation queue, to determine whether it is currently queued. If a queue exists, the count is reduced by one, the service time is computed, and an Evaluation Departure event is scheduled. However, if no queue exists, the server status changes to idle. Further, the system determines whether the pair of cars will pursue a lawsuit. If a lawsuit is filed ($C = 1$), a Complaining Arrival event is programmed; conversely, if no lawsuit is filed, a Fulfillment Arrival event is scheduled.

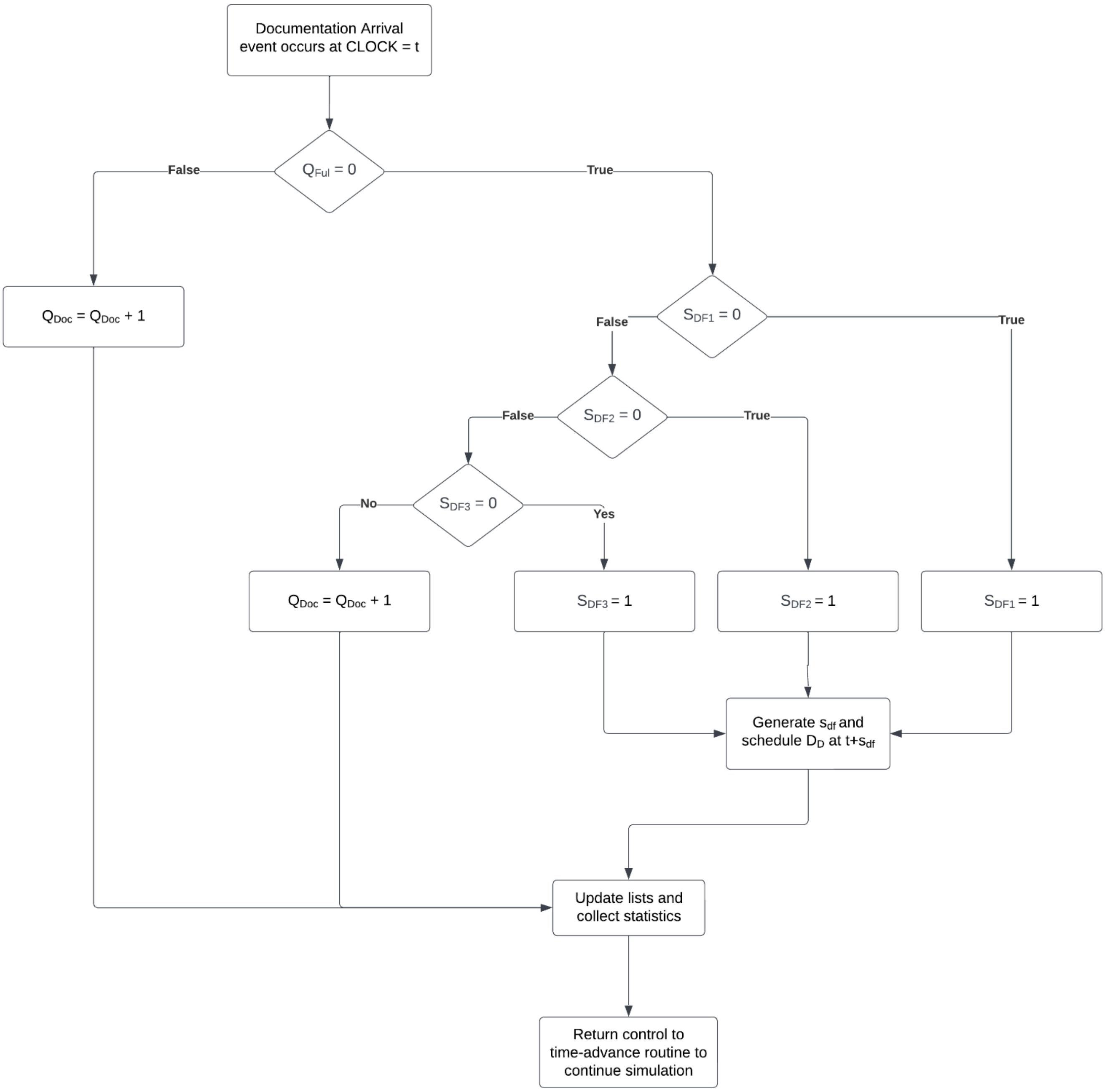

Figure 6- Documentation Arrival Event Flowchart

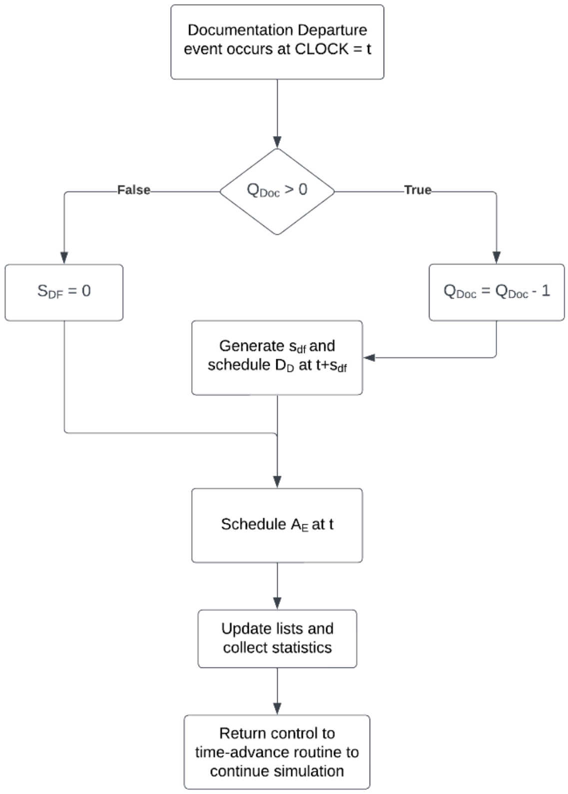

Figure 7- Documentation Departure Event Flowchart

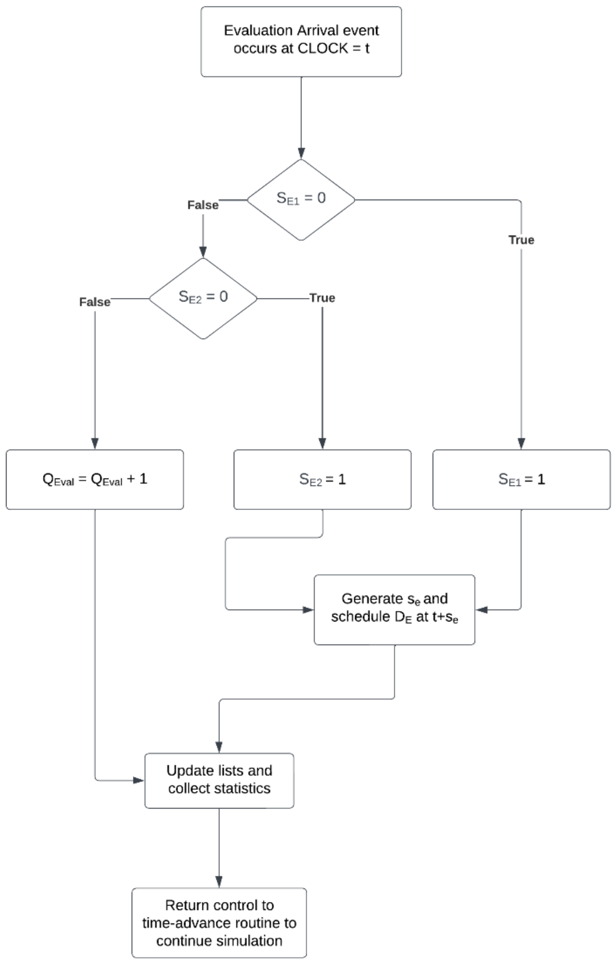

Figure 8- Evaluation Arrival Event Flowchart

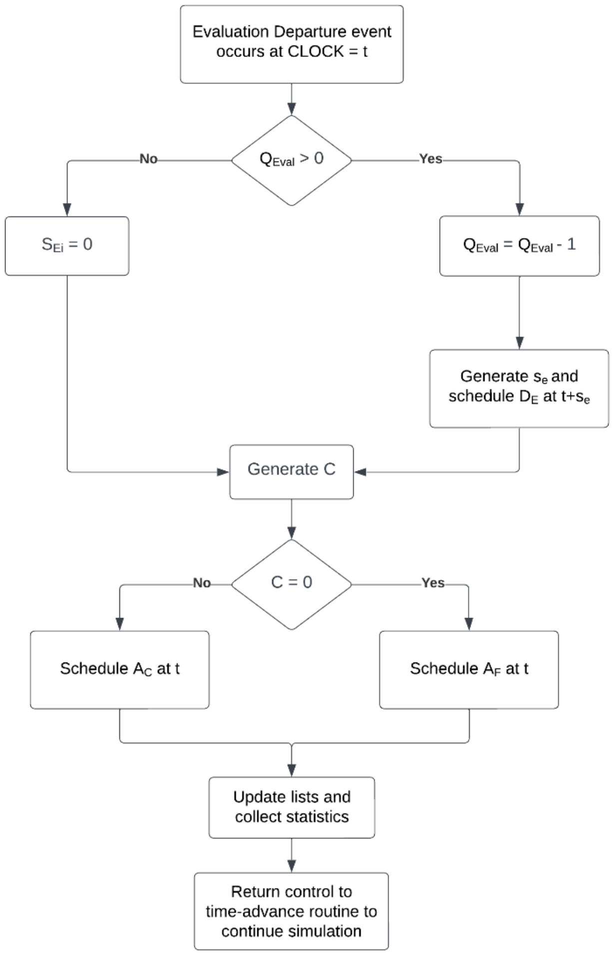

Figure 9- Evaluation Departure Event Flowchart

1.2.10 Complaining Arrival (A_C)

In this particular phase of the problem, a check is performed to ascertain the availability of the first specialist. If the specialist is idle, their status is updated to busy, and a service time is generated. Subsequently, the Complaining Departure event is scheduled. Alternatively, if the specialist is engaged, the pair of cars is added to the Complaining queue.

1.2.11 Complaining Departure (D_C)

In this section of the problem, an initial check is performed to determine if the Complaining queue is still active. In the case where a queue exists, the queue length is decremented by one, and the service time is calculated for scheduling a Complaining Departure event. Conversely, if there is no queue, the status of the servers transitions to idle. After these steps are completed, an evaluation arrival event is scheduled, as the clients will need to return to the center after filing a complaint.

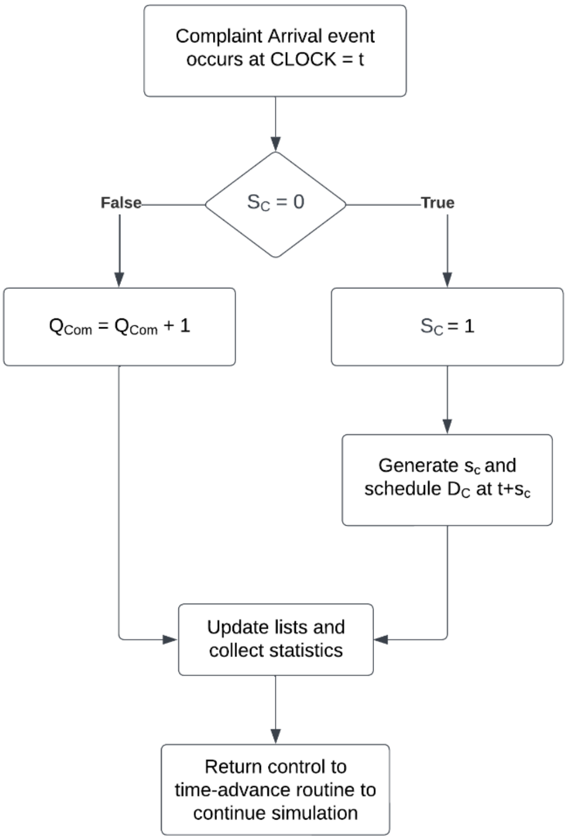

Figure 10- Complaining Arrival Event Flowchart

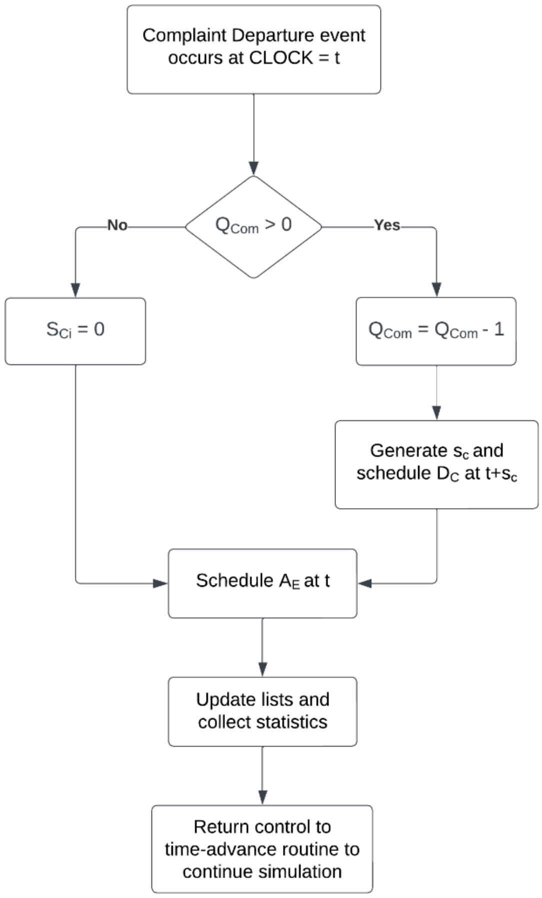

Figure 11- Complaining Departure Event Flowchart

1.2.12 Fulfillment Arrival (A_F)

In this particular phase of the problem, the system checks the availability of the first specialist. If the specialist is not busy, the system updates the specialist's status to "busy," creates a service time, and schedules the Fulfillment Departure event accordingly. If the first specialist is unavailable, the system repeats the same process for the second and third specialists. In the event that all three specialists are unavailable, the pair of cars is placed in the fulfillment queue. This process bears a striking resemblance to the documentation process as both occur within the same department.

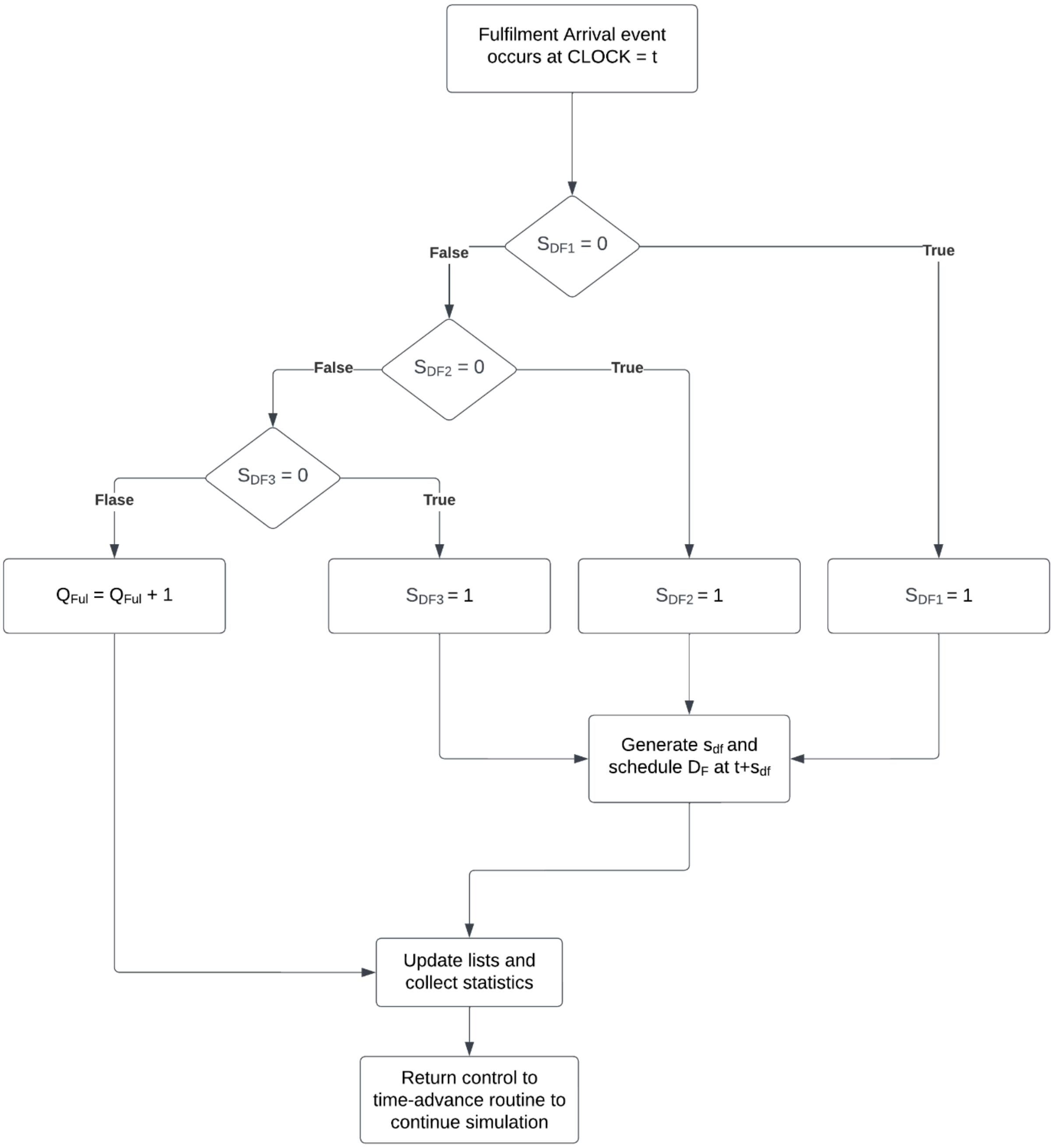

Figure 12- Fulfillment Arrival Event Flowchart

1.2.13 Fulfillment Departure (D_F)

In this section of the problem, the process for Fulfillment is initially checked to determine if a queue exists. If a queue is present, a decrement is made and the service time is calculated. A Fulfillment Departure event is then scheduled. In the absence of a queue, a check is made to determine if Documentation has a queue. If Documentation expert is idle, expert status is set as such. Alternatively, if a queue exists, a decrement is made, the service time is calculated, and a Documentation Departure event is scheduled.

Finally, a check is performed to determine whether the end of the shift has occurred and if all lists and queues are empty with all servers being idle. If these conditions are met, the time is reset to zero, and the entire process is repeated. If not, the simulation continues uninterrupted.

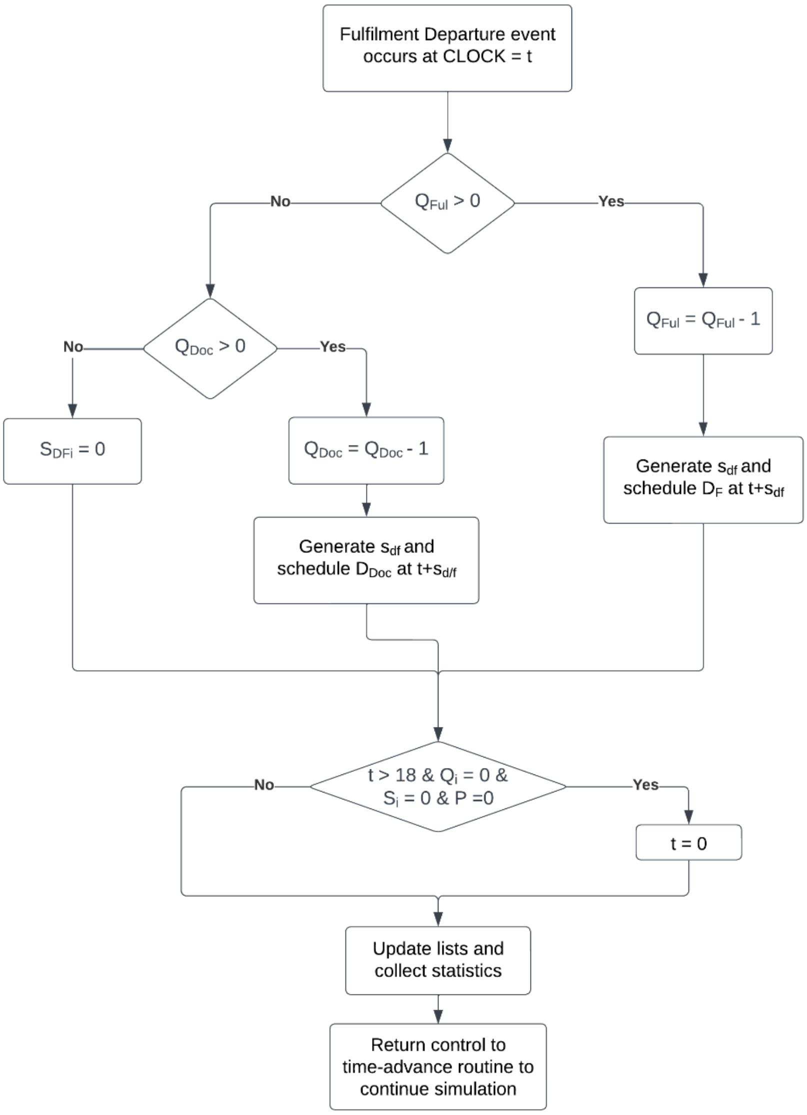

Figure 13- Fulfillment Departure Event Flowchart

1.2.14 Shif End (SE)

At the conclusion of the shift, precisely at time $t = 18$, the queue and list of individuals waiting outside the premises are reset to zero, effectively discontinuing their service. This process ensures that only individuals inside the premises continue to receive service until all are duly served.

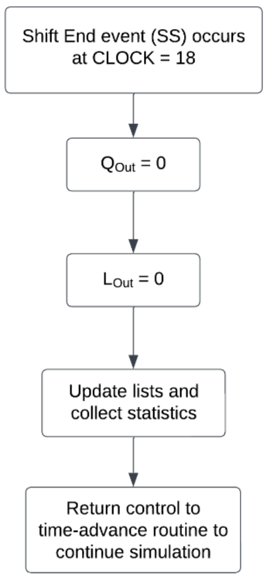

Figure 14- Shift End Event Flowchart

1.2.15 Performance Assessment Parameters

Average Service Time of Users: To estimate the number of users that can be served in a day, it is crucial to determine the average service time spent on each pair of clients. By calculating this value, we can better understand our current capacity.

Average Waiting Time of Users in Queues: Determining the average waiting time of pairs of cars in queues is important in assessing whether the number of users and experts is proportional. Dividing the total amount of time users have waited in queues by the number of users who entered the queues yields this value.

Average Time Users Are in the System: Clients will remember the amount of time they have spent on their problem. It is important to calculate the sum of the user's entry and exit time intervals and divide it by the number of users to determine the average time users are in the system.

Average Length of Each Queue: The length of the queue is a good indicator of the efficiency of the system. Calculating the time intervals with the same length of the queue and multiplying the length of the queue by the interval time, then adding this value for all the intervals provides the average length of each queue.

Maximum Length of Each Queue: Knowing the maximum number of users in each queue in the most critical situation is important for decision-making in managing the center. Calculating this value is done by finding the maximum value of Q throughout the simulation.

Productivity of Experts and the Whole System: One of the primary goals of simulation is to evaluate the efficiency of the system. Measuring the productivity of the experts and the entire system allows for planning measures to improve and increase efficiency.

***

Chapter Two; Phase 2 (Part 1)

2 Input Modeling

In this segment of the study, our objective is to identify the service distribution of servers based on the input data pertaining to the arrival and departure times of vehicles. To achieve this, we begin by preparing the data, and subsequently selecting a distribution that is both intuitive and in accordance with the nature of the phenomenon under investigation. Following this, we employ a q-q plot to visualize the distribution and conduct statistical tests such as the chi square and Kolmogorov-Smirnov tests to validate our hypothesis.

2.1 Data Preparation and Plotting

The given data includes a series of times which represent the start or the end of a service. For example, for three consecutive data entries $8:42$, $8:51$, $8:58$, the first and second one mark the start and the end of the first serving respectively, and the second and the third one shows the beginning and the end of the second serving, respectively. In order for the given data to mean anything, we had to turn them into values and thus, we extracted the intervals associated with them; these intervals would be the desired service times. Note that the data was collected from multiple days and the last rows of each day would be empty, which were removed using Microsoft Excel's filtering tool. This data is shown in the "Clean Data" column in the attached Excel document.

What we are left with are a series of service times in minutes. As instructed, a histogram of the data must be drawn, keeping in mind that the ranges (or bins) are required to have the same probability of being selected. Since we are given around 1000 entries ($n = 989$ valid service times), the number of bins must be in range of $\sqrt{n}$ and $n/5$; from which 32 was chosen and that leaves a bin length of 0.09.

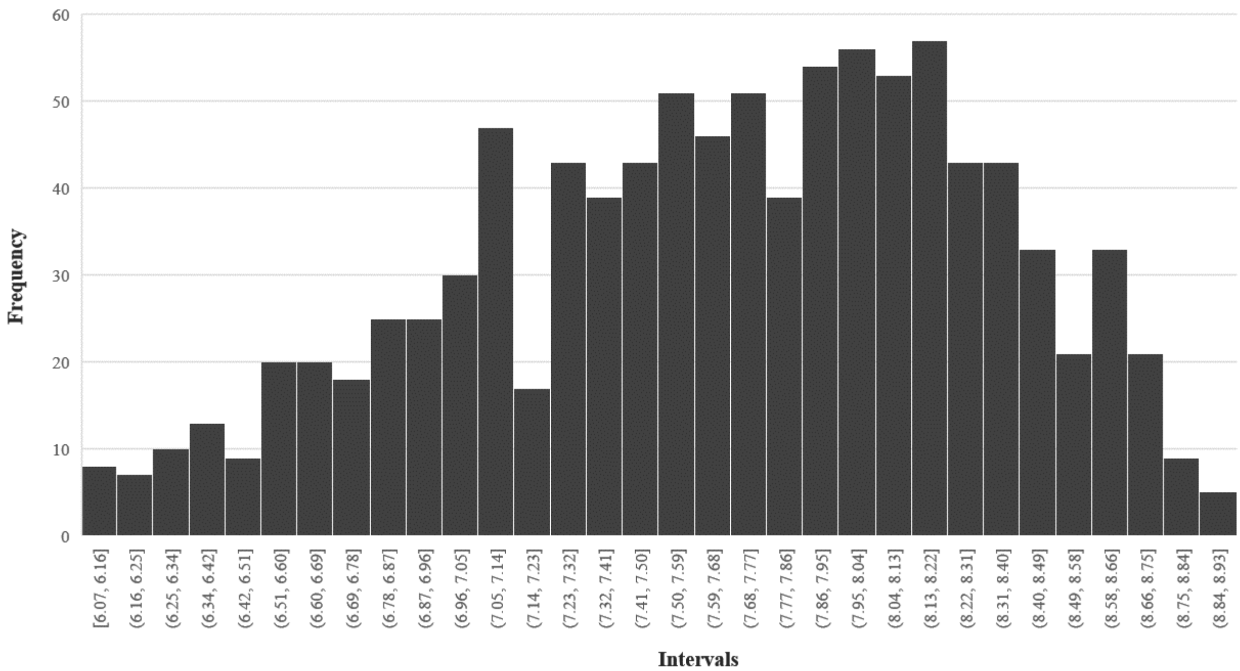

Figure 15- Histogram of Clean Data

2.2 Data Preparation and Plotting

Having drawn the histogram, it is now time to detect the data's distribution function. It is important to remember though, that we need to choose between the continuous functions because of the nature of values we are inspecting, which is *time*. It is observable that the distribution is slightly crooked to the right, which leads us to believe that normal and lognormal functions are out of the question. The only feasible option left is triangular distribution.

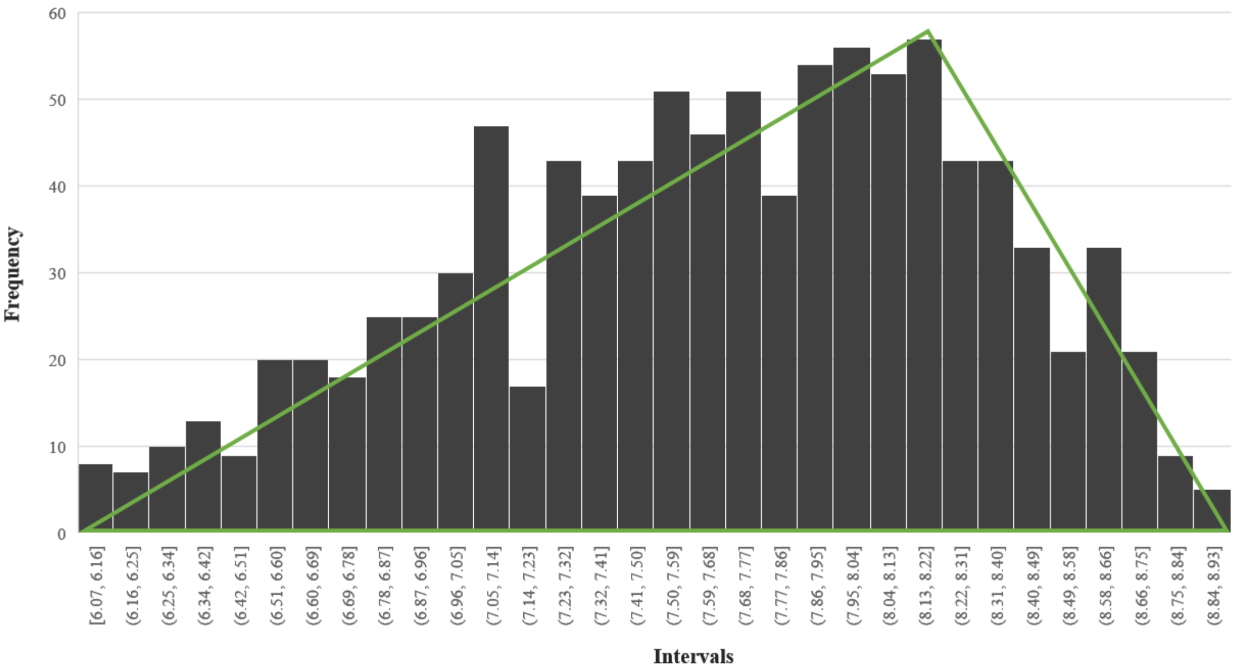

Figure 16-Triangular Distribution on the Data

After deducing the distribution function, parameters too must be estimated. These parameters are: $a = \min of\ data$, $b = \max of\ data$ and finally $c = mode\ of\ data$. As easy as it is to find the first two, finding the data's mode has to be done with caution; since it is always likely to wrongly estimate its value. In order to avoid mistakes, we can use the $(3 \times average) - min - max$ or other feasible values. We can even find the range containing the most data and use its mean as mode; we chose to do the latter. The parameters are found to be as follows:

$$a = 6.07$$

$$b = 8.93$$

$$c = 8.05$$

With the characteristics of the function discovered, we will move on to evaluate our hypothesis and determine whether the chosen distribution for the data is a good fit or not. We will do so in different ways such as a q-q plot, Kolmogorov-Smirnov, and Chi-square tests.

Prior to conducting such tests, it is imperative to precisely estimate the distribution and compute its cumulative distribution function. The subsequent section provides a comprehensive mathematical exposition of this crucial step. Notably, the equations derived in the ensuing section have been employed in the accompanying Excel document to determine the corresponding values.

2.3 Calculating CDF and Its Inverse

$$f(x) = \begin{cases} 0\,, & x < a \\ \dfrac{2(x-a)}{(b-a)(c-a)}\,, & a \le x < c \\ \dfrac{2(b-x)}{(b-a)(b-c)}\,, & c \le x \le b \\ 0\,, & x > b \end{cases} \quad \rightarrow \; F(x) = ?$$

$$for\ x < a : \int_{-\infty}^{t} 0dx = 0$$

$$for\ a \le x \le c : \int_{-\infty}^{a} 0dt + \int_{a}^{x} \frac{2(t-a)}{(b-a)(c-a)} dt = 0 + \frac{(t-a)^2}{(b-a)(c-a)} |_a^x = \frac{(x-a)^2}{(b-a)(c-a)}$$

$$for\ c \le x \le b : \int_{-\infty}^{a} 0dt + \int_{a}^{c} \frac{2(t-a)}{(b-a)(c-a)} dt + \int_{c}^{x} \frac{2(b-t)}{(b-a)(b-c)} dt$$

$$= 0 + \frac{(t-a)^2}{(b-a)(c-a)} |_a^c + \frac{(b-t)^2}{(b-a)(b-c)} |_c^x = 1 - \frac{(b-x)^2}{(b-a)(b-c)}$$

$$for\ x > b : \int_{-\infty}^{a} 0dt + \int_{a}^{c} \frac{2(t-a)}{(b-a)(c-a)} dt + \int_{c}^{b} \frac{2(b-t)}{(b-a)(b-c)} dt + \int_{b}^{\infty} 0dt$$

$$= 0 + \frac{(t-a)^2}{(b-a)(c-a)} |_a^c + \frac{(b-t)^2}{(b-a)(b-c)} |_c^b + 0 = 0 + \frac{c-a}{b-a} + \frac{b-c}{b-a} = 1$$

$$F(x) = \begin{cases} 0, & x < a \\ \dfrac{(x-a)^2}{(b-a)(c-a)}, & a \le x < c \\ 1 - \dfrac{(b-x)^2}{(b-a)(b-c)}, & c \le x < b \\ 1, & x \ge b \end{cases}$$

The next step forward is to find the inverse distribution function and set it as a_i, then solve for X and define it as a function of a_i. a_i is the quantile calculated as $(i - 0.5)/n$ and is fed to the inverse function using whose outputs, we can draw the q-q plot.

$$\rightarrow \quad X = \begin{cases} \sqrt{(b-a)(c-a)a_i} + a, & 0 \le a_i < \dfrac{c-a}{b-a} \\ b - \sqrt{(1-a_i)(b-a)(b-c)}, & \dfrac{c-a}{b-a} \le a_i \le 1 \end{cases}$$

The q-q plot is a good way of getting a notion of how accurate our estimation is. It is drawn using the actual data and the values generated by the inverse distribution function we previously defined. These two are compared in a scatter plot and if a straight trendline is observed, our choice can be interpreted as accurate.

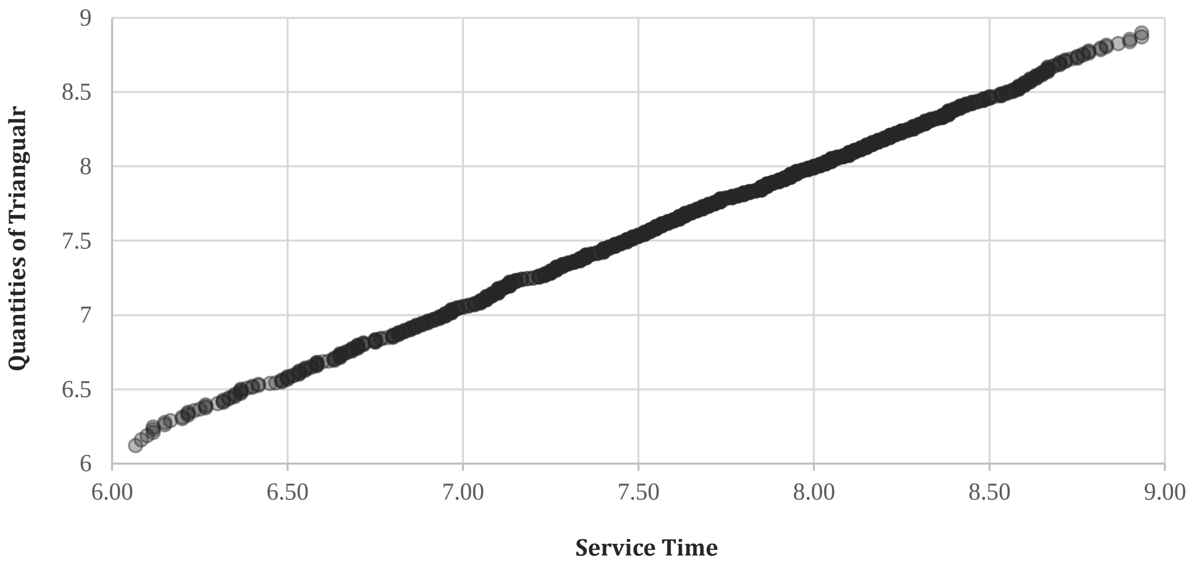

Figure 17- q-q plot for Triangular Distribution Hypothesis

It is visible from the plot that our estimation is adequately precise.

2.4 Goodness of Fit Tests

Goodness of fit tests are another step in finding out how the chosen distribution actually describes the given data. Two hypotheses are defined as our estimation being correct and it being incorrect. The result of the test will show us which we can accept and which we must reject. There are two tests of this kind as pointed out before, and one of them is usually enough for our goal; but we used both to ensure we are on the right track.

$$\begin{cases} H_0\!: \textit{The random variable is triangularly distributed with estimatetd parameters} \\ H_1\!: \textit{The random variable is not triangularly distributed with estimatetd parameters} \end{cases}$$

The first test is Chi-square. We define points in the span of our data between which the probability of selection is almost equal. This is done using the function's inverse and the input data is the cumulative probability of the ith row. The number of data that should be in each range is named E_i, and the number of ones that actually exist are named O_i. Also, the probability of selection is p_i and is calculated as one over number of ranges. Then the statistic is calculated and compared with the Chi Square statistic. If the statistic is lower than the calculated sum, the hypothesis of our estimation being accurate is accepted.

For the next test, Kolmogorov-Smirnoff, first the quantiles are i/n. These then enter the distribution's actual function. From this, two values, $D+$ and $D-$ are generated. $D+$ is the quantile minus the output of the same row's function, and $D-$ is the function's output minus the previous row's quantile value. The maximum value of $D+, D-$ are held against the test's statistics and like before, determine which hypothesis we can accept.

After analyzing the values obtained from both tests, we conclude that our null hypothesis, asserting that the data adheres to a triangular distribution, cannot be rejected. Detailed calculations for all the cases are available in the accompanying Excel document.

***

Chapter Three; Phase 2 (Part 2)

3 Python-based System Replication and Performance Analysis

To improve a system, it is essential to understand it. What it does, how it achieves its goals, what and how much resources it uses, and other aspects of its livelihood must be under careful watch. As it was obvious, measures were taken for this matter and a general image of our system took shape. The most important part however, is to build a replica as much accurate as possible in order to gain a valuable insight into the minor interactions that form the whole mechanism.

In this section, we implemented the model using Python to get live feedback from the system's performance. In our program, some functions were defined to target our varying needs throughout the programming. Then, the parts from which the data takes its inputs were created. These inputs include system parameters, data gathering and observation dimensions, and the ones related to the project's required outputs. In the last part, the simulation is run for the desired number of days and iterations and the data is collected and structured.

The simulation's code file has been provided as an attachment to the project. It includes comprehensive explanations of the arguments and the underlying logic to enhance clarity for the reader. The code is presented as separate blocks in the Jupyter Notebook environment, with each code block accompanied by an explanation block in markdown format. Furthermore, every line of code is thoroughly commented to facilitate the reader's understanding without encountering any difficulties. Consequently, in the forthcoming report, we shall refrain from delving into the intricacies of the code. Instead, our focus will be on presenting the challenges encountered and analyzing the desired outputs. In order to evaluate the code itself, it is advisable to open the provided .ipynb file using an appropriate IDE.

3.1 Coding Challenges

There were a lot of challenges in the way of successfully coding the whole system. The first one was defining the necessary functions and their area of applicability; since different parts of the code could have highly similar needs, it was difficult to determine exactly what each function did, and even whether or not it was necessary at all.

The second challenge was how to collect data from various system activities. This data was generated by users and their interactions with the system's components and very much varied in nature. Since we used lists to store them, the first approach of adding data to the lists was to use the *.append()* command. We soon came to realize though, that our lists cannot be treated like coin bags where added content just pile up without any particular order. What we did to overcome this challenge, was to create lists prepared and structured in advance, and assign desired data to the right place. We could then successfully track these lists' behavior and figure out the code's bugs much easier.

The third challenge was speed. We were aware that the process would create large amounts of data and working with them would be time consuming if not efficiently managed. For example, we needed to compare two datasets, the FEL and a file containing the queue lengths of different servers. Each of these contained about 90000 records and it was necessary to cross examine a certain type of data in both of them. That would require a loop within a loop which could take hours to operate. The solution was to replace one of the loops with a Pandas's built in function *.loc[]* which would allow us to put conditions in our search. This reduced the processing time to mere minutes.

The fourth setback was related to one of the project's requirements. It was pointed out that after 10 hours, the center will close down and only the customers who are already inside would continue receiving service. This would mean that no new customer could enter (which was easily implemented) and the queue formed outside would be disbanded. We tried different approaches and spent a lot of time to resolve this issue but the outside queue variable could not be emptied or in anyway affected. Whatever we did, it would hold its contents and continued to cause trouble. The solution was surprisingly simple though. By looking at the dataset created from the system's activities, we noticed that we could easily ignore the customers who arrived later than when the shift ended; in this way, no extra action would be needed and we were able to store the number of customers outside when the center closed down for the day.

The last major problem we faced was the differing nature of the FEL and some of the data we collected. The data that primarily went to the FEL was collected in discrete steps, because it was stored only when an action was carried out. These could include events like the customers' entry to the queues, when they started getting served, when they departed, etc. On the other hand, queue lengths, waiting times, and generally non sequential data could not be treated the same way. Instead, these were recorded in specific intervals (for example every 1 minute) and we were unable to directly connect them to the FEL data. The solution was to record queue lengths for as long as we could to be sure they encompassed all of the simulation lifespan, and link them to the nearest event in FEL. Using the sufficient amount of precision (like 0.2 minutes), this would create a complete and accurate FEL.

3.2 Output Analysis

After creating the program and running it with the desired characteristics, it is now time to extract the data and analyze it. The first dataset we are going to work with contains simulation's data grouped by their iteration number. Each of these iterations are a 30-day span (or whatever time period our user chooses) and the average amount of each of the factors is calculated. To grasp the overall system performance, a point estimation followed by a confidence interval is calculated for each of these factors. Point estimation is simply computing the 50 numbers' average and assuming it as the system's actual output amount.

As stated earlier, this approximate number is accompanied by an interval to show how much we can be certain that our estimate is accurate. In order to achieve this, we first choose a probability, like 95% (where α is 1 – 95% = 5%), to determine how much we need to be sure. Then, using an excel formula *CONFIDENCE.T(α, S, n)* or by manually calculating using $\frac{S}{\sqrt{N}} \times t_{\alpha/2,\ n-1}$, we will realize how much distance the ends of this interval have from the point estimation.

However, it is worth noting that our analysis has yielded a multitude of additional results and metrics beyond the scope of the project's requirements. By expanding our analysis to include a wider range of variables and measurements, we have gained a deeper understanding of the system under examination.

The full calculated values and data is represented in the attached .xlsx file called “Simulation Replications (98104111 - 98103891)”.

Let us take a look at the results. We can clearly see that by an acceptable estimate, about 250 pairs of cars are served by the center. Also, with not as much certainty, it seems like 150 pairs are lost every day because of the shift’s ending. Another valuable factor to assess the system’s overall performance by is how much time people spend in the whole establishment, which as we can infer from the data, is 440 minutes or more than 7 hours; which is a lot. Now if we wish to know what exactly is causing customers to stay in the system so long, we can refer to the times they spent queuing each server. By looking at these, we realize that although they did not wait a lot outside or in the complaint section, they did spend a shocking amount of time in the photography, documentation, and evaluation; nearly two hours in each. This is due to the large demand of these sections and the small number of servers in proportion.

Other important and not necessarily time-oriented factors exist too. We can view the efficiency of each server, and conclude that the complaint section is not really busy during the shifts. This is probably because only 10% of the people decide to file a complaint. It is noteworthy that other servers are working at near full capacity. We are also able to see the lengths of each queue. These numbers tell us that, the number of people waiting outside is actually higher than those inside. This is actually intriguing because as you might recall, customers did not spend much time in the center’s exterior.

There are also other parameters to extract meaningful data from, and one can dedicate a lot of time to it, observing, assessing, gaining insight, and finally make decisions in order to improve the system’s overall performance and work out its various chokepoints and issues; this mode offers it all.

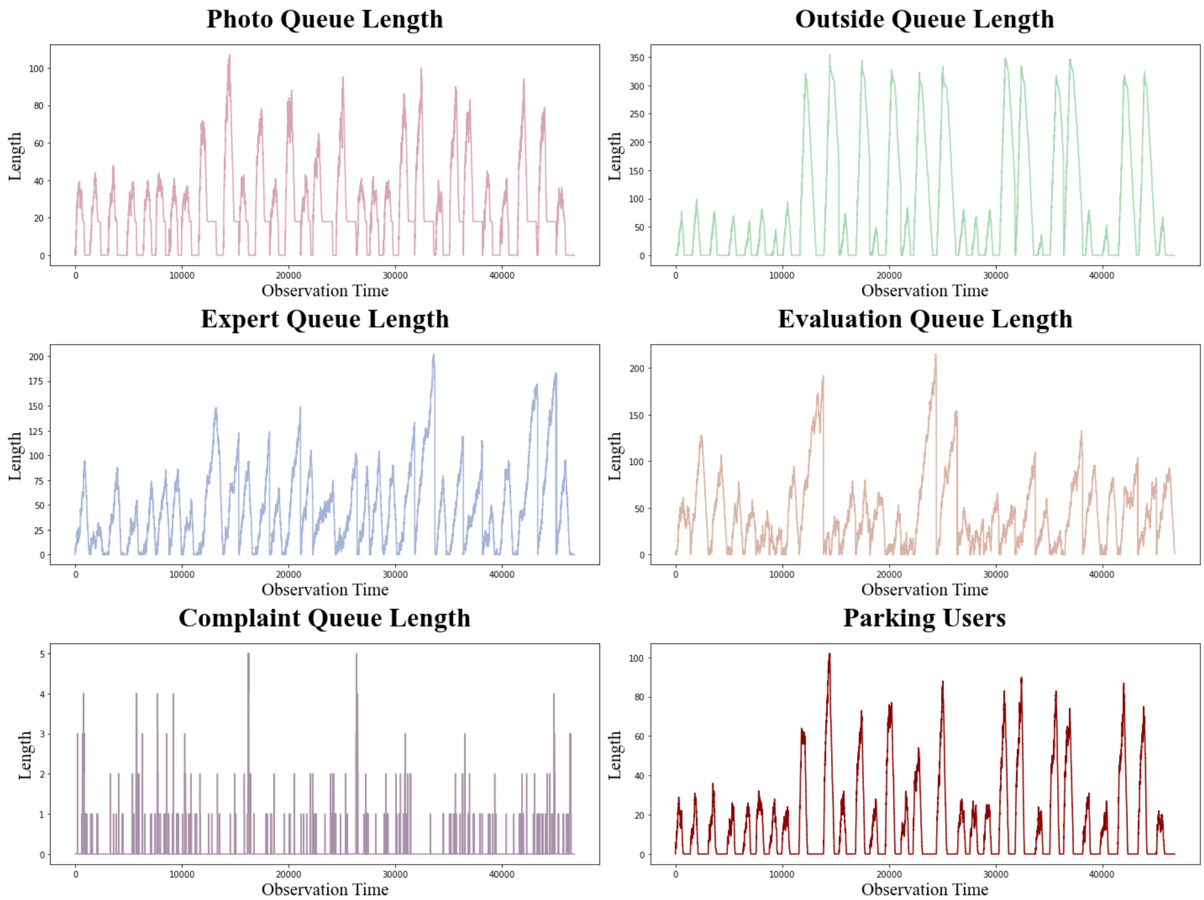

Figure 18- Server queues and parking usage by time

In addition to analyzing output numbers, valuable insights can be gained from charts and viewing trends. Here we are provided with six charts, depicting changes in servers' queue lengths and the number of customers waiting for their partner in the parking lot. These indicate that the system has generally followed the same pattern, with the servers' queue lengths peaking a couple of times a day at various numbers, from a not-so-busy server with a maximum queue size of 4, to the super busy section of evaluation that peaks at 350 users for multiple times in a day in we can clearly see the effect of having few employees. Here we are able to conclude what decisions should be made for the whole system; determining parking space, assigning more workers to critical points, or even hire a less qualified employee to cut down on costs.

3.3 Sensitivity Analysis

Towards the conclusion, we shall undertake a concise sensitivity analysis to further elucidate the significance of these operational models. Given that all inputs in this model are specified as variables within our coded program, conducting the sensitivity analysis becomes a straightforward process of manipulating individual values and observing corresponding outputs. In light of this, by altering selected input variables and contrasting the resulting outputs with those obtained in the preceding section, we can gauge the impact and relevance of the operational models in a more precise manner.

In this context, within the same Simulation Replication file, we have conducted a sensitivity analysis of the output data and presented point estimates using three different fundamental approaches. Firstly, we deliberately increased the entry time for cars by a factor of ten, as compared to the initial scenario. It is reasonable to expect a significant decrease in the number of serviced cars and other associated variables under such conditions, and this expectation was clearly observed.

Furthermore, another examination revealed that experts are predominantly occupied with both documentation and fulfillment tasks. Consequently, we modified the expert count from 3 to 6, with the anticipation that their efficiency, as an indicator of their workload, would diminish. This anticipated effect was evident in the output data as well.

Lastly, we observed that the duration of time spent in the Photography queue was excessively long. Thus, by reducing the photography service time, we expected a considerable reduction in this duration and an increase in the number of serviced cars. As anticipated, this adjustment resulted in logical changes across all variables.

Overall, these findings demonstrate the logical and expected outcomes of the alterations made in the simulation analysis.

***

Chapter Four; Phase 3

4 Simulation and Performance Assessment of Two System Alternatives

As pointed out before, simulation helps us grasp a system's performance in real-life situations with outstanding precision. The valuable insight this method helps us achieve is also useful in deciding on how to change and modify the system in order to get the best results. Using simulation, we can analyze the available options and put them through various environments. This will aide us see possible enhancements as clearly as we can see our existing system, and after careful consideration, we are able to test our hypothesis without much cost and act fast to exploit different windows of opportunity.

In this final phase, we were presented with two alternative systems to replace the old one. In each of these a number of changes took place in order to improve the center's overall performance. System 1 sees the parking lot section completely removed and operates under the assumption that all customers arrive in pairs. Also, the center is modified to accept customers 24/7 with an exponential distribution whose mean is 5. System 2 is a revised version of system 1 in which customers' arrival probability has a mean of 3.2 instead of 5 and the complaint section has been moved to another location. In addition, 4 new experts were hired and replaced whose serving time follows a triangular distribution with parameters 6, 8, and 10 in the documentation sector and 3, 3.5, and 4 in fulfillment. A new Evaluation employee was also hired and with some training, the mean serving time for experts in this section was lowered to 8 minutes.

We are required to implement these systems as we did for the previous one and choose the best option to replace the old one. For this matter, we follow the exact steps as we did before.[4]

To facilitate a systematic evaluation and comparison of the proposed alternatives, the code has been organized into three distinct *.ipynb* files. The first two files, aptly labeled "*Project_Phase3_System1_98104111_98103891*" and "*Project_Phase3_System2_98104111_98103891*" represent the meticulously coded models of the respective alternative systems, incorporating all the aforementioned enhancements.

Conversely, the third file, named "*Project_Phase3_System1_vs_System2_98104111_98103891*" serves the crucial purpose of executing final comparisons on the output generated by each system. All outputs are thoughtfully saved as *.CSV* files, thereby enabling the storage of crucial information such as queue lengths, server efficiency, and queuing dynamics, among other relevant parameters.

It is noteworthy that both systems undergo execution for an equal duration of time and the same number of replications, ensuring fair and balanced assessments. Each simulation run spans a period of 6 months, equivalent to 259200 minutes, thereby providing a comprehensive view of the systems' performance over an extended timeframe. Furthermore, to enhance the statistical validity and reliability of the comparisons, 100 replications are conducted for each system. This

[4] For detailed explanations and walkthroughs for the Python code, it is highly recommended to refer to the comments and explanations embedded in the relevant files.

considerable number of replications ensures that the simulation outcomes are based on a robust sample size, reducing the impact of random variations and strengthening the significance of the results.

By conducting these simulations with consistent timeframes and an adequate number of replications, the comparison between the two alternative systems is based on sound and reliable data, thus facilitating more accurate insights into their respective performances. Moreover, the generation of illustrative charts through the Python codes allows for a visual representation of key performance metrics, aiding in a more informed decision-making process for selecting the optimal system to replace the old one.

4.1 Warm-Up Period Analysis

After extracting the desired outputs from the code, we must determine when the system reaches its steady state and when the warm-up period concludes. In these periods, the system takes some time at first to adjust itself and reach the steady state. It is important to know exactly when each server or resource reaches this state and what number it stays on.

The significance of the truncation point becomes evident in this context, as it is essential for conducting comparisons and analysis between the two systems using steady state data. This specific point is determined by the user or analyst when they are presented with charts displaying queue lengths over time for various servers. They are then asked to identify the time before which the data is deemed to belong to the warm-up period and therefore, is considered irrelevant for the analysis. It is important to note that the data size before the truncation point should be at most one-tenth of the remaining data for accurate and precise analysis.

4.1.1 Methodology

As previously mentioned, each system underwent a continuous 6-month run, comprising 100 replications. At regular intervals of two minutes, various indicators and system status were meticulously examined and recorded. Consequently, a vast amount of data, approximately 130,000 data points for each metric and index, was amassed for every repetition. To be more precise, the cumulative dataset for each index across these 100 replications amounted to an impressive 13 million data points.

To mitigate potential errors and fluctuations stemming from probable distributions and to enhance the accuracy of the outputs, a prudent step was taken to compute the average data across all 100 repetitions pertaining to each individual observation as calculated below (where "i" represents the specific observation step, and "r" denotes the repetition number). This averaging process not only aids in ensuring the robustness of the results but also serves to alleviate any undue influence from individual replication variances.

$$\bar{Y}_i = \frac{\sum_{r=1}^{100} Y_{ir}}{100}$$

This averaged data for all servers during the specified time period was then plotted. To facilitate a clearer understanding of the data's change process, a moving average with a 300 data window and a polynomial regression with a high degree were performed on each dataset. Subsequently, the decision-maker or user should carefully examine the process and status of all servers and determine the point at which the latest server has achieved its steady state. This identified truncation point ensures that the simulation reaches a steady state, thus enhancing the robustness and reliability of the subsequent data analysis.

4.1.2 Results and Findings

The following display showcases the results derived from the aforementioned procedures for the first system.

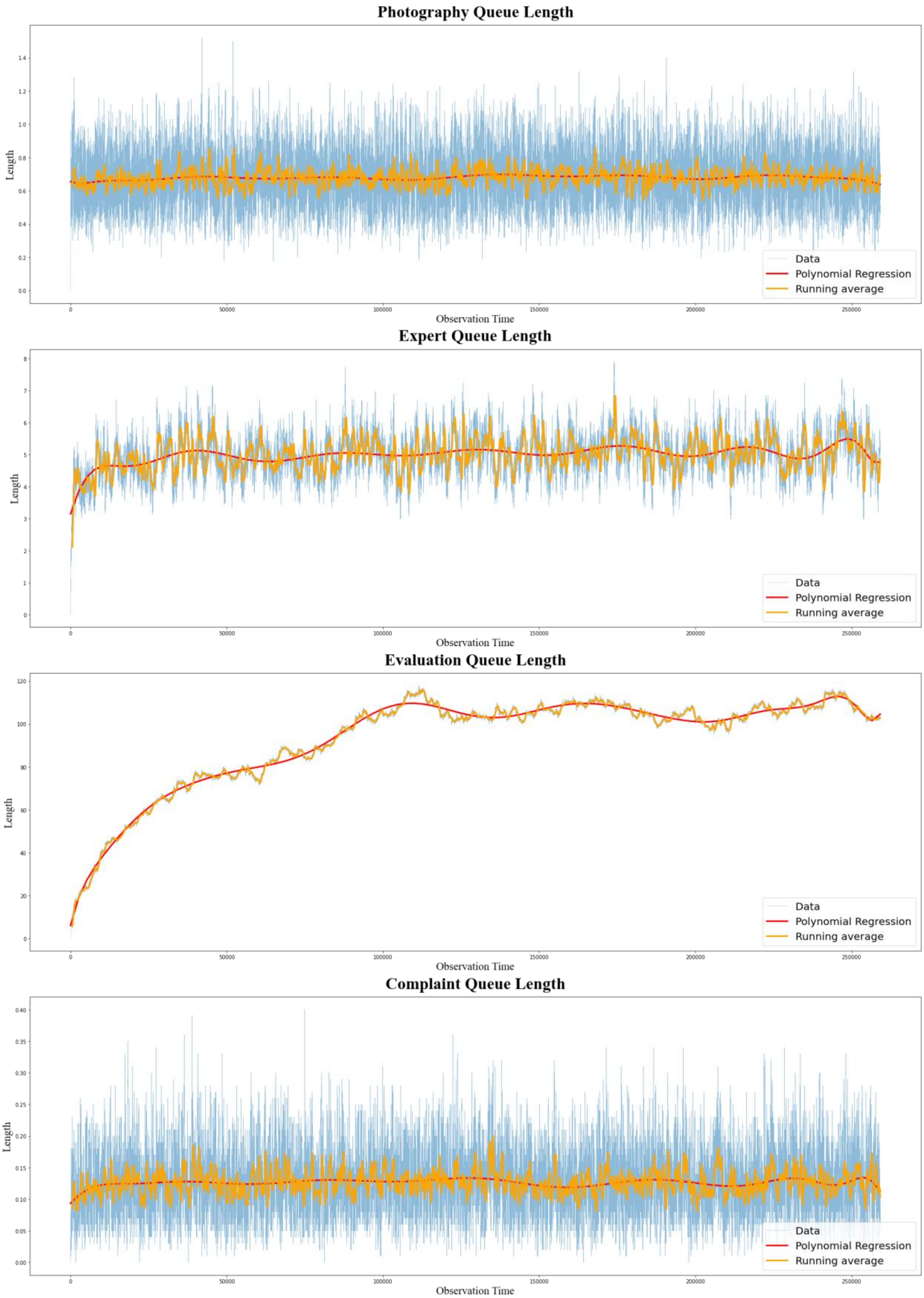

Figure 19- System 1's long-run performance

As evident from the data, the photography queue reached a state of stability early in the simulation, almost at its outset, with an average queue length of 0.65 individuals (a value considered small and highly desirable). This outcome can be attributed to the relatively longer time it takes for customers to enter the photography department compared to the pace of service provided.

Subsequently, the complaint queue also achieved stability, settling at approximately 0.2 individuals after approximately 5000 minutes. The rapid convergence and minimal size of this queue can be attributed to the significantly lower influx of customers, estimated to be less than 10%, combined with a considerably higher service rate compared to the entry rate.

In contrast, the expert queue took a longer duration of about 25,000 minutes to reach stability, maintaining an average of 5 people. This delayed stabilization can be explained by the department's dual responsibility for documentation and fulfillment, leading to a higher likelihood of conflicts arising.

However, the evaluation section displayed not only a much later attainment of stability but also a notably longer queue length. This queue started to heat up after approximately 80,000 minutes, accommodating an average of 110 people. The primary cause of this observation, however, is apparent: the evaluation section acts as the bottleneck in the entire process, as all customers pass through it, and those with complaints go through it twice. Moreover, the average service time for this section closely matches the arrival rate, leading to a prolonged period of increasing queue length and ultimately reaching a high level of stability.

Based on the findings, it is prudent to select the 80,000th minute as the truncation point for the initial system setup to ensure that all servers have undergone sufficient warming up, accounting for the varying dynamics of each section.

Earlier, we discussed the preference for retaining a dataset that is tenfold larger than the discarded data following truncation. Nonetheless, due to the delayed stabilization of the evaluation queue, the identification of an optimal truncation point based on this criterion becomes unfeasible. It is worth emphasizing that the system's significance during this particular stage outweighs the sheer volume of data relative to the data that has been removed.

Now we undertake a similar procedure to select the warm period for the second system. The trend of server queues in this system is depicted in figure 20. Notably, the complaint section is no longer present, leaving us with the need to examine only three other servers.

Various alterations in the input data have given rise to behavioral changes in this system when compared to its predecessor. Specifically, an increase in the rate of customer arrivals, resulting in reduced average time between arrivals, has led to the photography section's queue reaching a stable state at a later time (approximately 30,000 minutes) and accommodating around 12.5 individuals.

Contrasting with the photography section, changes in the average service times in the documentation and fulfillment sections, with no alteration in the queue size (remaining at 5 people), have resulted in reaching a stable state much earlier, approximately after 10,000 minutes.

However, the most significant transformation lies in the evaluation section. The addition of a new expert to this department and the improvement of their response speed have remarkably stabilized this server after approximately 30,000 minutes, with merely 3 people in the queue. This achievement stands as a substantial improvement compared to the previous system.

Based on the aforementioned instances, the 30,000th minute emerges as the truncation point for this system, indicating that the introduced changes and enhancements have expedited its warming process.

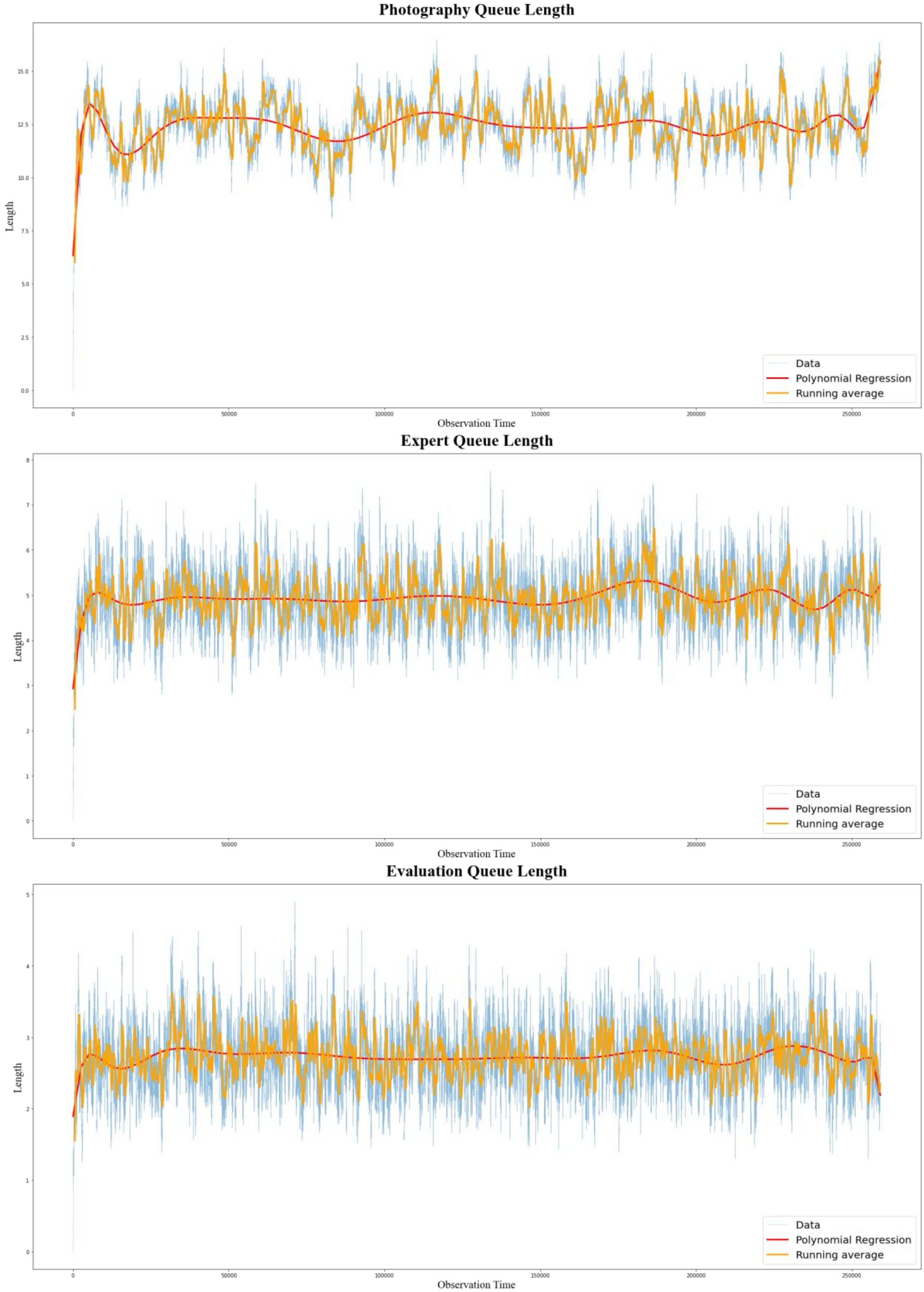

Figure 20- System 2's long-run performance

4.2 Comparative Study of Two Systems

Upon establishing structured datasets for both systems, relevant performance factors are gathered, and statistical measures, such as point estimation of their overall averages, confidence intervals, and variance, are computed. These data constitute the basis for making an informed decision among the available alternatives. The resulting output files are denoted as "System 1 Output Data" and "System 2 Output Data," respectively.

As previously stated, the comparison between these two systems employs an independent sampling approach. This selection method was deemed appropriate due to the absence of identical sections and flows that would enable the synchronization of seeds for both entities. Hence, independent sampling allows for a more feasible and valid comparison between the two systems. Subsequently, employing the t-test and computations pertaining to the comparative analysis of two systems, we assessed them from the vantage point of 28 distinct indicators. In accomplishing this, we conducted a subtraction operation by deducting the values pertaining to system two from their corresponding counterparts in system one ($\bar{Y}_{System\ 1} - \bar{Y}_{System\ 2}$).

It is crucial to acknowledge that the comparison conducted through the independent sampling method is based on the underlying assumption that, for the factors being examined, higher values are deemed preferable, signified by the confidence interval lying entirely above zero. Although this assumption holds true for certain aspects, such as servers' efficiency and the probability of queues being empty, it may not be universally applicable across all factors. More specifically, in the context of factors such as queue length, customers' overall queuing experience, and the time spent in specific sections, lower values are actually regarded as more desirable in an academic and practical sense, signified by the confidence interval lying entirely below zero.

The computation of confidence intervals for each factor is conducted as follows:

$$Point\ Estimate\colon\ \bar{Y}_1 - \bar{Y}_2$$

$$Standard\ Error\colon\ s.e.(\bar{Y}_1 - \bar{Y}_2) = \sqrt{\frac{S_1}{R_1} + \frac{S_2}{R_2}}$$

$$Degrees\ of\ Freedom\ (v)\colon\ \frac{\left(S_1^2/R_1 + S_2^2/R_2\right)}{\left[\frac{\left(S_1^2/R_1\right)^2}{R_1 - 1} + \frac{\left(S_2^2/R_2\right)^2}{R_2 - 1}\right]^2}$$

$$Confidence\ Interval\ for\ \theta_1 - \theta_2\colon\ \bar{Y}_1 - \bar{Y}_2 \pm t_{\alpha/2,v} \times s.e.(\bar{Y}_1 - \bar{Y}_2)$$

The performance details of the systems across various factors, along with the superior alternative, are illustrated in Figure 21. Our method employs a t interval approach using data collected from each system as shown above. When the interval encompasses zero, it suggests that the performance of both options in that specific area is statistically indifferent. However, if the t interval does not include zero, we can confidently assert, with acceptable precision, that one system outperforms the other. Furthermore, the magnitude of the interval's deviation from zero is significant as it indicates the extent of the gap between the two alternatives.

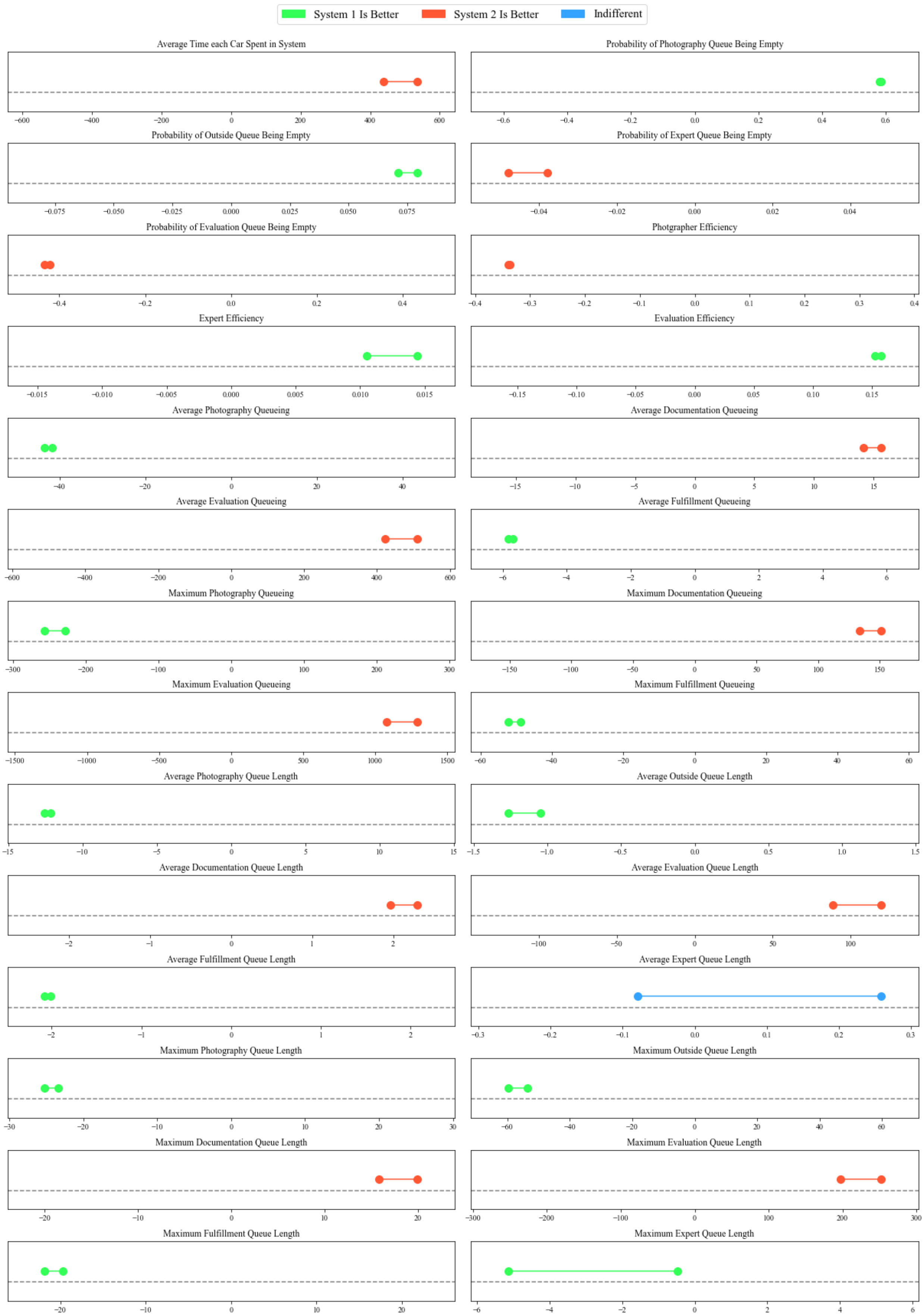

Figure 21- Confidence intervals for various factors comparing two systems

As evident from the analysis, the comparison of these systems based on different indicators has yielded diverse outcomes. For instance, the confidence interval pertaining to the *Average Time each Car Spent in the System* lies entirely above zero, indicating a significant difference between the systems in this aspect. Considering that a lower value is more favorable for this factor and we obtained the difference by subtracting the values of system 2 from system 1, a positive confidence interval implies that, on average, cars tend to spend more time in system 1, which is considered undesirable. Consequently, system 2 outperforms system 1 in this index, as denoted by its depiction in red to signify this result.

To provide further clarity on the implications of this section and Figure 21, let us consider another example: the confidence interval of *Expert Efficiency* is entirely positive, as indicated by the fourth confidence interval from the top left. However, unlike the previous indicator, a higher value in this index indicates better performance. Hence, it can be concluded that system 1 exhibits superior performance from this perspective, as denoted by its depiction in green to signify this result.

Moreover, during the analysis of the warm-up period, it was observed that the number of people in the expert queue, both in the long term and after the system had warmed up, remained relatively consistent, at around 5. Consequently, it was anticipated that a definitive conclusion regarding the comparison of the two systems in this index might not be attainable, and this turned out to be the case. The confidence interval of the *Average Expert Queue Length* index encompasses the value of zero, rendering it inconclusive and preventing the determination of one system's superiority over the other, as denoted by its depiction in blue.

To facilitate comprehension and enhance comparability between these two systems, we have also constructed the following table.

Table 1- Comprehensive comparison between system 1 and system 2

Factor	**System 1 Is Better**	**System 2 Is Better**
Average Time each Car Spent in System	-	✓
Probability of Photography Queue Being Empty	✓	-
Probability of Outside Queue Being Empty	✓	-
Probability of Expert Queue Being Empty	-	✓
Probability of Evaluation Queue Being Empty	-	✓
Photgrapher Efficiency	-	✓
Expert Efficiency	✓	-
Evaluation Efficiency	✓	-
Average Photography Queueing	✓	-
Average Documentation Queueing	-	✓
Average Evaluation Queueing	-	✓
Average Fulfillment Queueing	✓	-
Maximum Photography Queueing	✓	-

Table 1- Comprehensive comparison between system 1 and system 2 (Cont.)

Factor	System 1 Is Better	System 2 Is Better
Maximum Documentation Queueing	-	✓
Maximum Evaluation Queueing	-	✓
Maximum Fulfillment Queueing	✓	-
Average Photography Queue Length	✓	-
Average Outside Queue Length	✓	-
Average Documentation Queue Length	-	✓
Average Evaluation Queue Length	-	✓
Average Fulfillment Queue Length	✓	-
Average Expert Queue Length	-	-
Maximum Photography Queue Length	✓	-
Maximum Outside Queue Length	✓	-
Maximum Documentation Queue Length	-	✓
Maximum Evaluation Queue Length	-	✓
Maximum Fulfillment Queue Length	✓	-
Maximum Expert Queue Length	✓	-
Overall Assessment	**15**	**12**

It is evident that the first system has exhibited superior performance across a broader range of indicators, albeit with only marginal differences. Notably, the second system outperforms in the realms of documentation and evaluation. However, when considering the domains of photography, fulfillment, and even off-site parking, the first system demonstrates a more commendable performance.

Thus, the selection of the most suitable system ultimately hinges on the subjective judgment and preferences of the decision-maker, as well as the relative importance assigned to each of the aforementioned indicators. As a prospective avenue for future research, a comprehensive evaluation could involve the assignment of weights to these indicators and subsequent application of the Analytic Hierarchy Process (AHP) for a more rigorous and informed decision-making process.

4.3 Further Study

Through simulation, we are able to effectively assess various systems, identify areas for improvement, and facilitate iterative enhancements. This approach enables us to examine the outcomes of diverse actions and compare different strategies without directly impacting the actual operational system. By saving valuable time and resources, we aim to deliver an enhanced customer experience, ensuring their satisfaction and loyalty.

Despite the potential benefits of simulation, it is evident that real-world scenarios pose challenges. Insurance queues persistently remain lengthy, and reform policies often encounter implementation issues, resulting in ineffective outcomes. The underlying reasons for these complexities can be multifaceted, encompassing various factors such as:

- Implementation issues with reform policies leading to ineffective outcomes: When attempting to implement new policies or reforms, various obstacles can arise that hinder their effectiveness. These issues may stem from resistance to change, bureaucratic complexities, or unforeseen consequences.
- Inherent unpredictability of human behavior and decision-making: Human behavior can be difficult to predict accurately, and decision-making processes may vary from person to person, making it challenging to model and simulate accurately.
- Limited data availability or accuracy: Simulation models heavily rely on data to make accurate predictions and assessments. In some cases, the available data may be insufficient or inaccurate, leading to less reliable simulation outcomes.
- Inadequate modeling of certain real-world intricacies: Real-world systems can be highly complex, and some intricacies may be challenging to capture adequately in simulation models, leading to potential inaccuracies.
- Unforeseen interactions between different components of the systems being simulated: In complex systems, interactions between various components can lead to unexpected outcomes that may not be apparent in simulations, causing discrepancies between predicted and actual results.
- External factors such as changes in regulations, economic conditions, or technological advancements that may impact the effectiveness of simulation-based assessments: The real world is dynamic, and external factors can significantly influence the performance of simulated systems. Changes in regulations, economic fluctuations, or advancements in technology can all affect the validity of simulation results.

Addressing these challenges is crucial for organizations aiming to leverage simulation effectively in decision-making processes. While simulations offer a controlled and cost-effective environment for testing and refining systems, the gap between simulation outcomes and real-world results can hinder their utility. To harness the full potential of simulations, it is essential to recognize the intricacies of real-world complexities and refine the simulation models accordingly. Moreover, understanding the limitations of simulations and the factors contributing to discrepancies between simulations and reality is essential for devising appropriate solutions. By acknowledging these challenges and actively working towards mitigating them, organizations can unlock the true value of simulation in driving innovation, optimizing processes, and enhancing customer experiences. Now, let's proceed with the ways to solve these challenges:

- Continuous refinement of simulation models: To improve the accuracy and reliability of simulations, models need to be continuously updated and enhanced based on new data and insights.
- Comprehensive data gathering: Gathering comprehensive and high-quality data is essential for developing accurate simulation models and understanding real-world complexities better.

- A deep understanding of the real-world context in which the simulations are applied: Context is crucial for simulations to be meaningful and relevant. Understanding the real-world environment and the specific issues being addressed ensures simulations are tailored to the right scenarios.
- Diligent effort and constant adaptation to improve customer experiences and drive better outcomes: Overcoming the challenges of real-world simulations requires ongoing commitment, adaptability, and a willingness to learn from both successes and failures to deliver enhanced customer experiences and more effective results.

As part of this study, based on our observations of the results, we present some conclusive recommendations aimed at enhancing the efficacy of this center:

- Addressing the issue of burgeoning workloads on evaluation experts and general personnel responsible for documentation and fulfillment is crucial. To alleviate their burden and streamline processes, we propose augmenting the workforce by recruiting additional personnel and redistributing responsibilities more evenly among them.
- Implementing comprehensive training programs tailored to the specific needs of each department's employees can be instrumental in reducing the average service time within these units. By expediting customer flow through the system, such training initiatives can effectively curtail waiting periods.
- A noteworthy solution involves bifurcating the documentation and fulfillment departments and providing specialized training to the respective experts in their designated domains. This approach serves to mitigate the potential bottleneck that could arise within these critical areas.

By implementing these recommendations, the center stands to optimize its overall operational efficiency and enhance the quality of service provided to its clientele.

***

Chapter Five; Conclusion

In this academic project, a discrete-event systems simulation approach was applied to analyze and optimize the performance of a car insurance assessment center. The project was divided into three phases, each contributing crucial insights into the system's dynamics and potential improvements.

Phase One involved defining the problem statement and providing a static description besides a dynamic one of the center. This initial groundwork helped understand the system's challenges, including limited capacity, priority rules for car arrivals, and queue length constraints.

Phase Two comprised two parts focusing on input modeling and implementation using Python. Data preparation and statistical tests were conducted in Part 1 to validate the distribution of service times for servers. The triangular distribution was identified as the most suitable choice. In Part 2, the simulation model was implemented using Python, considering various factors such as queue lengths and server efficiency.

Phase Three marked the simulation and performance assessment of two alternative systems aimed at enhancing the center's efficiency. Warm-up period analysis determined the stabilization points for queues and sections. A comparative study of the alternative systems was conducted, showcasing System 1's superiority in photography and fulfillment, and System 2's better performance in documentation and evaluation.

This project highlighted the power of simulation as a decision-making tool for optimizing complex systems. By replicating real-life scenarios and evaluating a wide range of performance indicators, simulation provided valuable insights into the system's behavior. The study emphasizes the significance of careful data analysis and model building to ensure accurate and reliable simulation results. Overall, this academic endeavor contributes to the understanding and improvement of real-world systems, enabling businesses and organizations to enhance their operational efficiency and customer satisfaction through informed decision-making.