Drive-Thru Dynamics: A Simulation-Based Analysis of Fast Food Operations

Project 5, Group 8: Zubair Lakhia

1 Abstract

The drive-thru has become an important aspect of the fast food industry that offers custonmers both speed and convenience. However, sometimes long wait times and service bottlenecks remain persistent especially during peak hours. The following paper goes over a discrete-event simulation model to help analyze the dynamics of a fast-food drive through system on a small scale. The simulation has a three stage process which covers process ordering, payment and pickup while considering factors such as service/station configurations/customer arrival rates etc. The model was built using SimPy a simulation framework in Python. Results from the simulation reveal that the pickup station was the biggest issue, with wait times significantly reduced by adding more pickup stations. The results demonstrate that choosing resources wisely and system configurations can improve wait times and service reliability.

2 Introduction

The invention of the drive-thru in 1930's was revolutionary. It has become an integral component of the fast food industry and is a convenient way for customers to obtain meals without leaving their vehicles. The drive-thru is a staple for many large national chains such as McDonald's, Wendy's, and Burger King. Drive-thrus thrived in the COVID-19 pandemic when businesses relied heavily on these operations to minimize in person contact and adhere to social distancing guidelines. During this time drive-thrus usage was noted to significantly increasee with estimates suggesting almost a 43% rise in usage in 2020 [1]. These services offered a critical solution to food industry revenue but they also exposed various inefficiencies, especially in managing long wait times and optimizing service delivery [2].The long queues can lead to customer dissatisfaction, and could lead to some customers abandoning orders and turning to competitors. These problems would often arise during peak lunch hours or when the queue was inefficient in general. To address these challenges, many businesses have started to utilize simulation based models to analyze and improve their drive-thru performance. Simulation models can incorporate factors such as customer arrival rates, service times, and employee configurations. Customizing these factors can identify bottlenecks and test various wait time strategies. Previous studies have shown that such as tablets, order boards, and other types of assistive technology have sped up service times, but the overall impact on service quality and operational efficiency is still difficult to quantify [1]. With simulation based tools, businesses can easily evaluate various approaches, such as adjusting employee count, optimizing service lanes or even changing the layout of their drive-thrus to ensure the highest possible revenue. Understanding the processes that agents go through within a system is an important component of logistical planning, especially for large scale organizations. This paper aims to develop a simple yet effective simulation-based tool to analyze drive-thru service dynamics. The tool will allow for the testing of different employee and customer scenarios which could potentially be then used to provide valuable insights into how resources can be better allocated to minimize waiting times.

3 Proposed Method

The study employs a discrete-event simulation model to investigate the efficiency of a fast-food drive-through system. Before using SimPy (Simulation in Python), extensive background research was completed through various articles and sources found on the internet, which reviewed examples of other simulations created in SimPy. This was done to gain familiarity with the general framework and obtain ideas on how to construct the drive-through model. After thorough background research, the plan was to use the drive-through model to analyze the impact of key operational variables such as order stations, payment stations, customer arrival rates, and average wait times and queue lengths. The goal was to identify any bottlenecks and optimize system performance under different configurations.

3.1 Simulation Model

The simulation models a three-stage drive-through system:

(1) **Ordering** - Customers place orders at one of the available order stations.
(2) **Payment** - Customers proceed to a payment station to complete their transactions.
(3) **Pickup** - Customers receive their orders at the pickup station before exiting.

The data was artificially generated to model a realistic fast-food drive-through based on a local chic-filet near my house. This synthetic data generation allows for structured analysis of system performance under different conditions. Each station has a limited number of service points (resources), and customers queue when all stations are occupied. Service times at each stage follow a uniform random distribution within predefined time ranges, while inter-arrival times follow an exponential distribution, simulating real-world customer arrivals.

3.2 Performance Evaluation

To evaluate system performance, the parameters were modified with each run, and analytics were measured. The simulation runs for 120 minutes, with customer wait times, queue lengths, and service utilization rates recorded.

3.3 Initialized Parameters

```
RANDOM_SEED = 42
NUM_ORDER_STATIONS = 4
NUM_PAYMENT_STATIONS = 2
NUM_PICKUP_STATIONS = 1
ARRIVAL_RATE = 5
SERVICE_TIME_ORDER = (1, 3)
SERVICE_TIME_PAYMENT = (1, 2)
SERVICE_TIME_PICKUP = (2, 5)
SIM_TIME = 120
```

3.4 Data Collection Parameters

- **wait_times** → Tracks how long customers spend in the system.
- **queue_lengths** → Records queue sizes at each station.
- **service_times** → Stores the actual time customers spend at each stage.
- **station_utilization** → Tracks station usage levels.

Each customer moves through the order → payment → pickup steps. In the ordering step the customer requests access to an order station and if all stations are busy, the customer waits in line. Once a station is available, the customer spends 1 to 3 minutes ordering. The time spent is recorded and added to the queue. In the payment step, the customer also waits if necessary before proceeding. In the initial test run with stationary parameters the pickup step is usually the bottleneck since only one station is available. The customer waits for an available pickup station and then receives their order. The total time spent in the system is recorded. The customer generator function randomly generates customers using an exponential distribution to simulate realistic arrival patterns. Two graphs to represent total wait times and queue length trends are shown below.

4 Results

The results of the drive-thru simulation model are shown below. First, analytics of the stationary parameters were measured, followed by a more robust simulation with varying parameters.

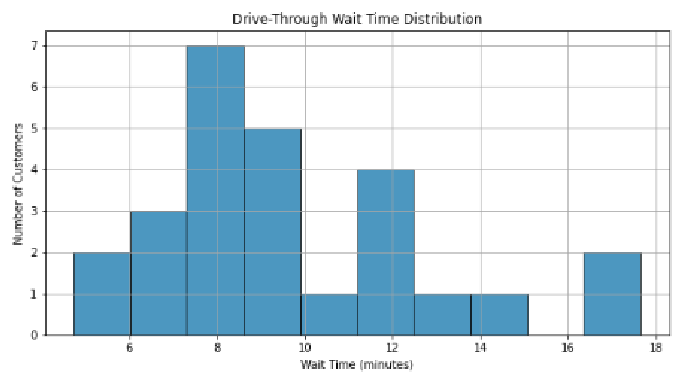

Figure 1

Figure 1 shows the distribution of total wait times that customers experience in the drive-thru simulation model for the stationary parameters model. We can see that most customers wait between 6-12 minutes and that the peak is around 8 minutes. Some customers wait longer, suggesting that the bottlenecks could be at the pickup stations. The distribution is right skewed and the longer tail indicated that a subset of customers get stuck in the system longer, possibly due to congestion at specific stations.

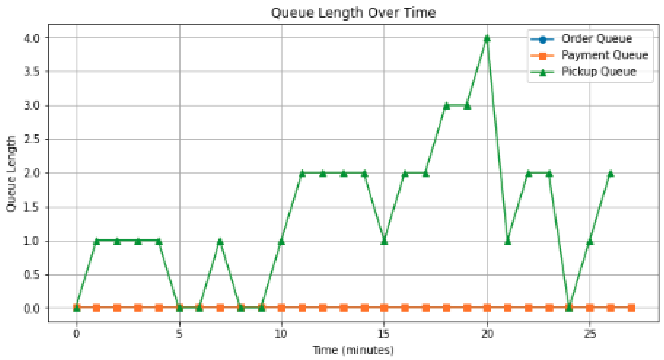

Figure 2

Figure 2 shows the queue lengths at each stage of the drive-thru system over the course of the stationary parameter simulation model. The pickup queue seems to fluctuate significantly and build over time and peaks around 3-4 customers waiting at once. The payment queue is constantly low which means that the payment stations are efficient and never have major delays and the same logic applies to the order queue which is almost not visible. Customers are flowing smoothly through the system until they reach the pickup stage.

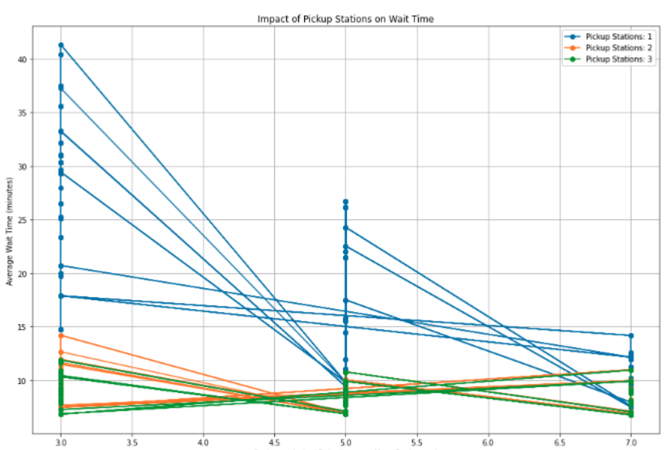

Figure 3

Figure 3 is a graph representing the modified and varying parameters for the next drive-thru simulation model. This graph illustrates how average wait times are affected by the number of pickup stations as customer arrival rates change. Some important observations are that blue lines (1 pickup station) show the longest wait times, sometimes exceeding 40 minutes. Orange lines (2 pickup stations) show significant improvement, keeping wait times below 15 minutes. Green lines (3 pickup stations) have the shortest wait times, mostly below 10 minutes. Based off this, we can see that more pickup stations reduce wait time. Another key takeaway is that higher arrival rate (More Customers) does not always increase wait time. As the arrival rate increases (moving right on the graph), wait times decrease slightly or stabilize for higher pickup station counts. This suggests that with more customers, the system balances itself because idle times at service stations decrease. The blue lines (1 pickup station) fluctuate heavily, meaning some simulations produced very long wait times, while others were more reasonable. This can suggest that with only 1 pickup station, the system becomes unstable, leading to unpredictable wait times. 2-3 pickup stations seem to stability wait times due to the orange (2 pickup stations) and green (3 pickup stations) lines being much more stable and clustered together.

Best (Lowest Wait Time) Configurations:

	Order Stations	Payment Stations	Pickup Stations	Arrival Rate \
136	3	2	3	7
352	5	2	3	7
568	7	2	3	7
208	3	3	3	7
424	5	3	3	7

	Order Time	Payment Time	Pickup Time	Avg Wait Time
136	(1, 3)	(1, 2)	(2, 5)	6.740944
352	(1, 3)	(1, 2)	(2, 5)	6.740944
568	(1, 3)	(1, 2)	(2, 5)	6.740944
208	(1, 3)	(1, 2)	(2, 5)	6.758958
424	(1, 3)	(1, 2)	(2, 5)	6.758958

Worst (Highest Wait Time) Configurations:

	Order Stations	Payment Stations	Pickup Stations	Arrival Rate \
367	5	3	1	3
583	7	3	1	3
77	3	2	1	3
149	3	3	1	3
293	5	2	1	3

	Order Time	Payment Time	Pickup Time	Avg Wait Time
367	(2, 4)	(2, 3)	(3, 6)	41.312134
583	(2, 4)	(2, 3)	(3, 6)	41.312134
77	(2, 4)	(1, 2)	(3, 6)	40.366360
149	(2, 4)	(1, 2)	(3, 6)	40.366360
293	(2, 4)	(1, 2)	(3, 6)	37.454914

Figure 4

Figure 4 shows a table displaying the best (lowest wait time) and worst (highest wait time) configurations for the fast-food drive-through simulation.

5 Conclusion

To wrap things up, the small scaled simulation-based analysis described above of drive-thru operations has proven to be a valuable tool in understanding and improving the efficiency of fast food systems. By employing a discrete-event simulation model, this study identified bottlenecks in the system, particularly at the pickup station, where the long queues and extended wait times caused significant delays. The findings suggest that increasing the number of pickup stations significantly reduces wait times and stabilizes the system, especially during high customer arrival rates. Also adding a few more service points helped with congestion showing that if you place your resources wisely you can really optimize overall service efficiency. Moving forward, we can add more variables to the model to incorporate real world data and more complex service configurations.

References

[1] Liam Whitenack and Ron Mahabir. 2022. A Tool for Optimizing the Efficiency of Drive-Thru Services. In *2022 Systems and Information Engineering Design Symposium (SIEDS)*. 151–156. https://doi.org/10.1109/SIEDS55548.2022.9815014 Accessed: 2025-02-04.

[2] Nuru Ain Athirah Zulkarnain and Azila Md Sudin. 2024. Optimization of Waiting Time at Drive Thru Fast Food Restaurant using Queuing Model. *Enhanced Knowledge in Sciences and Technology* 4 (Dec. 2024), 134–140. Issue 2. https://doi.org/10.1234/ekt.2024.0132

- https://medium.com/@noel.B/a-complete-guide-to-using-simpy-for-ai-simulations-testing-fdc4ed1cf271#:~:text=SimPy%20is%20an%20excellent%20framework,projects%20requiring%20detailed%20environment%20modeling
- https://realpython.com/simpy-simulating-with-python/ (Heavily used for framework)
- https://simpy.readthedocs.io/
- https://docs.python.org/3/tutorial/datastructures.html
- https://campus.datacamp.com/courses/discrete-event-simulation-in-python/developing-discrete-event-models-using-simpy?ex=5
- https://stackoverflow.com/questions/70937689/plotting-multiple-line-graphs-in-matplotlib
- https://matplotlib.org/stable/index.html
- https://stackoverflow.com/questions/30030659/in-python-what-is-the-difference-between-random-uniform-and-random-random#:~:text=105-,random.,end%20up%20giving%20you%20b%20)
- https://www.geeksforgeeks.org/sympy-stats-uniform-in-python/
- https://stackoverflow.com/questions/69609682/how-to-set-initial-queue-length-in-simpy