Queue Simulation: Testing and Performance Analysis

Erfan Bayat
s328390
erfan.bayat@studenti.polito.it

***Abstract*—This report presents a discrete event simulation model for analyzing multi-server queuing systems with priority scheduling and preemption capabilities. The model implements different queue disciplines and incorporates client classification, allowing for the evaluation of service quality metrics under various operational conditions. Results demonstrate the system's ability to handle complex queuing scenarios while maintaining detailed performance metrics.**

I. Introduction

This report presents a multiserver queue simulation with configurable server count and queue capacity. The objective is to evaluate the impact of different service policies—First Come, First Served (FCFS), Last Come First Served (LIFO), and a Priority-based system with preemptive and non-preemptive configurations—on system performance.

II. Methodology

the simulation was built with two primary parameters

A. System Parameters

- **server count:** Configurable to simulate different levels of system capacity.
- **Queue Capacity:** Specifies the maximum number of customers that can wait in line. Once capacity is reached, additional arrivals are dropped.

B. Service Policies

- **First Come, First Served (FCFS):** Customers are served in the order they arrive.
- **Last Come, First Served (LIFO):** Newly arriving customers are prioritized over those already in the queue.
- **Priority-based:** Customers are assigned urgency levels, with higher-priority customers being served first. This policy has two configurations:
  - **Preemptive:** Higher-priority arrivals interrupt ongoing service for lower-priority customers. (in our simulations only the highest priority has the preemptive privilege)
  - **Non-preemptive:** Higher-priority arrivals do not interrupt ongoing service.

III. Simulation Setup

This section outlines the parameters used to configure the queuing system simulation. These parameters determine the behavior and constraints of the simulated environment.

- **Number of Servers (c):** 3 servers are used to handle incoming customers, allowing for concurrent service processing.
- **Queue Capacity (K):** The queue can hold up to 5 customers. Any customer arriving when the queue is full is considered rejected.
- **Arrival Rate (λ):** 0.8 customers per unit time, indicating an average of one arrival every 1.25 time units.
- **Service Rate (μ):** 0.6 services per unit time, meaning each server completes a service on average every 1.67 time units.
- **Simulation Time (T):** The simulation runs for a total of 1000 time units.

IV. Performance Metrics

The simulation records various performance metrics to assess the efficiency and responsiveness of the queuing system under different service policies. The key metrics are as follows:

Total Customers Arrived,Total Customers Served, Rejected Customers, Preemption, Average Delay, Standard Deviation of Delay, Average Queue Size, Loss/Dropping Probability and Remaining Customers

Additional priority-specific metrics are recorded for each customer type (e.g., VIP, Regular, and Economy) to further evaluate performance under priority-based policies: Customers Arrived by Priority, Customers Served by Priority Average Delay by Priority and Standard Deviation of Delay by Priority

V. Simulation Results

This section presents the performance results for each service policy (FCFS, LIFO, and Priority-based) implemented in the simulation. The metrics used to evaluate each policy include average delay, standard deviation of delay, average queue size, and loss probability.

A. FCFS Policy Results

The simulation utilized a FIFO (First In, First Out) queue with 3 servers and a queue capacity of 5. A total of 805 customers arrived, out of which 803 were served, and 1 customer was rejected. At the end of the simulation, 1 customer remained in the queue. The average delay experienced by customers was 0.90 time units, with a standard deviation of 1.46 time units.

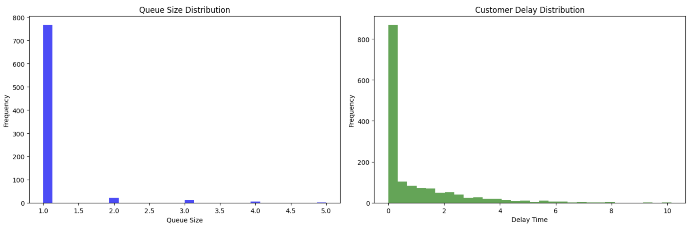


Fig. 1: Queue Size Distribution for FCFS Policy

B. LIFO Policy Results

The simulation utilized a LIFO (Last In, First Out) queue with 3 servers and a queue capacity of 5. A total of 821 customers arrived, out of which 820 were served, and only 1 customer was rejected. At the end of the simulation, there were no remaining customers. The average delay experienced by customers was 1.01 time units, with a standard deviation of 1.60 time units.

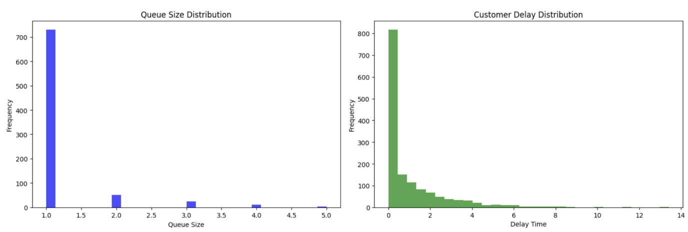


Fig. 2: Queue Size Distribution for LIFO Policy

C. Non-Preemptive Priority Results

This section presents the results for the priority queue without preemption. To better visualize the system's impact, the service rate was reduced to 0.3 in sections C and D.

1) Customer Statistics: The simulation used a non-preemptive priority queue with 3 servers and a queue capacity of 5. Of the 766 total customers, 734 were served, 28 were rejected, and 4 remained in the queue at the simulation's end. The average queue length was 1.96, and the average waiting time in the queue was 4.51 time units.

2) Priority-Specific Statistics:

- **VIP Customers:** 159 arrived, all of whom were served, with an average delay of 4.13 time units.
- **Regular Customers:** 364 arrived, with 362 served, resulting in an average delay of 5.38 time units.
- **Economy Customers:** 215 arrived, with 213 served, experiencing an average delay of 8.60 time units.

D. Preemptive Priority Results

This section presents the results of the preemptive priority queue simulation.

1) Customer Statistics: The simulation utilized a preemptive priority queue with 3 servers and a queue capacity of 5. Out of a total of 766 arriving customers, 729 were served, 33 were rejected, and 4 remained in the queue at the simulation's end. The average queue length was 1.99, with an average waiting time of 4.55 time units. There were 143 instances of preemption throughout the simulation.

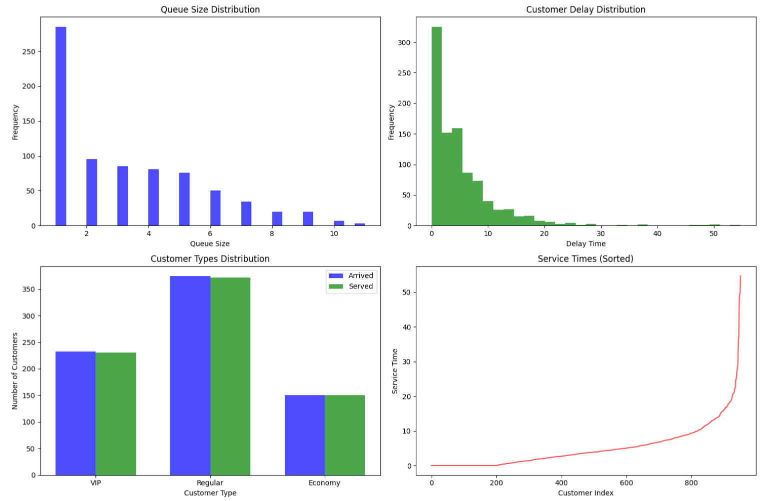


Fig. 3: Queue Size Distribution for Non-Preemptive Priority Policy

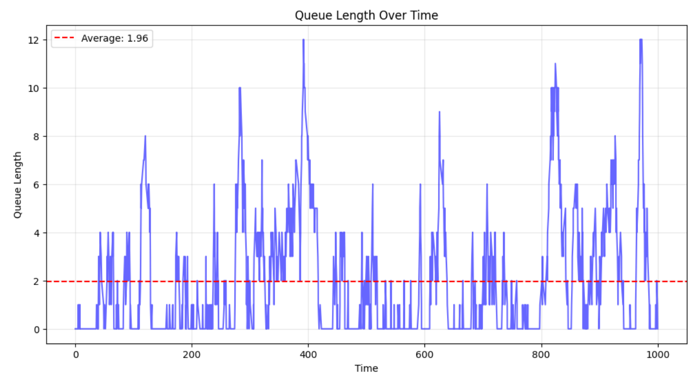


Fig. 4: Queue length over time for non-preemptive case

2) Priority-Specific Statistics:

- **VIP Customers:** 159 arrived, with all 159 served. The average delay experienced was 3.05 time units.
- **Regular Customers:** 365 arrived, with 358 served, resulting in an average delay of 5.51 time units.
- **Economy Customers:** 214 arrived, with 212 served, with an average delay of 9.01 time units.

based on the delay time we can confirm the preemptive is in fact working

E. Comparison of Simulation Results with Theoretical Values

In this section, we compare the performance metrics obtained from the simulation of an M/M/1 queue with the corresponding theoretical values. The parameters for our simulation and the theoretical model are as follows:

- **Queue Type:** FIFO Queue
- **Number of Servers:** 1
- **Queue Capacity:** Infinite

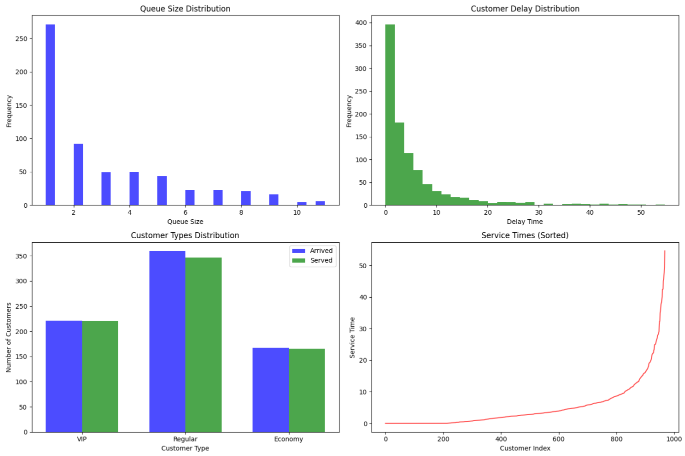


Fig. 5: Queue Size Distribution for Preemptive Priority Policy

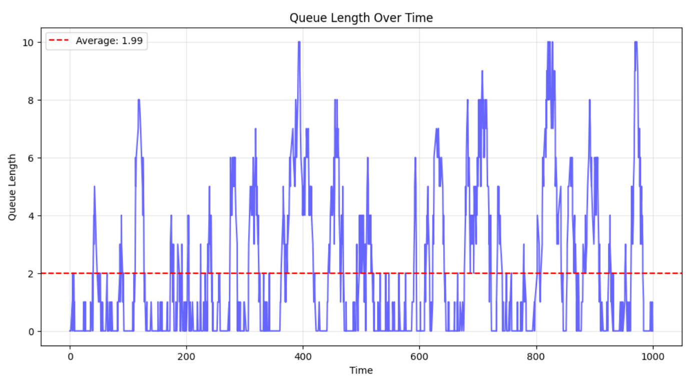


Fig. 6: Queue length over time for preemptive case

The performance metrics from the simulation are summarized below:

- **Average Queue Length** (L_q)**:** 0.39
- **Average Waiting Time in Queue** (W_q)**:** 0.51 time units
- **Average Time in System** (W)**:** 1.76 time units

The theoretical values for the M/M/1 queue are provided as follows:

- **Utilization** (ρ)**:** 0.5
- **Average Number of Customers in the System** (L)**:** 1.0
- **Average Time in the System** (W)**:** 2.0 time units
- **Average Number of Customers in the Queue** (L_q)**:** 0.5
- **Average Time in the Queue** (W_q)**:** 1.0 time units

1) Analysis of Results: The simulation results show that the average queue length (L_q) is 0.39, which is slightly lower than the theoretical value of 0.5. This indicates that the simulated queue was less congested than expected in the theoretical model.

For the average waiting time in the queue (W_q), the simulated value of 0.51 time units is also lower than the theoretical value of 1.0 time units. This suggests that customers experienced shorter waiting times in the simulated environment compared to the theoretical predictions.

The average time in the system (W) was found to be 1.76 time units in the simulation, which is less than the theoretical average of 2.0 time units. This discrepancy may be attributed to variations in arrival and service patterns during the simulation.

Overall, while the simulation results show a fair approximation of the theoretical values, they reveal a more efficient queue performance than anticipated, indicating a potential need for further analysis of the simulation parameters and their impact on queue dynamics.

VI. Summary

In this report, we presented a discrete event simulation model designed to analyze multi-server queuing systems under various service policies, including FCFS, LIFO, and priority-based configurations. The model allows for the evaluation of system performance metrics such as average delay, queue size, and rejection rates, while incorporating features like client classification and preemption capabilities. The simulation results indicate that different service policies significantly impact system performance, with the priority-based strategies showing distinct advantages in serving high-priority customers. Furthermore, comparisons between simulated results and theoretical values for an M/M/1 queue suggest that our simulation model demonstrates a more efficient queue performance than the theoretical predictions. These findings underscore the importance of configuring queuing systems to optimize service quality and operational efficiency.

Gaussian Random Variable Generation and Testing

Erfan Bayat
S328390
erfan.bayat@studenti.polito.it

***Abstract*—This report presents an in-depth exploration of methods for generating Gaussian random variables using Newton's method for the inverse-transform technique, the Box-Muller transform, and the Central Limit Theorem (CLT). We compare the performance of these methods by evaluating their distribution accuracy and sample independence through comprehensive statistical tests, including chi-square, Kolmogorov-Smirnov (K-S), autocorrelation, and runs tests.**

I. Introduction

This report compares three methods for generating Gaussian random variables:

- Inverse-transform technique using Newton's method
- Box-Muller transform
- Central Limit Theorem (CLT) method

Each method will be tested for accuracy in reproducing the $N(0,1)$ distribution and independence between samples.

II. Methodology

A. Inverse-Transform Technique with Newton's Method

The inverse-transform technique generates Gaussian random variables by inverting the cumulative distribution function (CDF) of the standard normal distribution. The CDF, $F(x)$, is given by:

$$F(x) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{1}$$

Where erf is the error function. To generate a random variable X such that $X \sim N(0,1)$, we take a uniform random variable $U \sim U(0,1)$ and compute:

$$X = F^{-1}(U) \tag{2}$$

We use Newton's method to numerically approximate the inverse of the CDF:

$$x_{n+1} = x_n - \frac{F(x_n) - U}{f(x_n)} \tag{3}$$

where $f(x)$ is the PDF of the standard normal distribution.

B. Box-Muller Transform

The Box-Muller transform generates pairs of independent standard normal random variables from two independent uniform random variables. Let $U_1, U_2 \sim U(0,1)$. The transformation is given by:

$$Z_1 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2) \tag{4}$$

$$Z_2 = \sqrt{-2\ln(U_1)}\sin(2\pi U_2) \tag{5}$$

C. Central Limit Theorem (CLT)

The Central Limit Theorem (CLT) states that the sum of a large number of independent, identically distributed (i.i.d.) random variables, each with finite mean and variance, will be approximately normally distributed. To generate Gaussian random variables, we take the sum of n uniform random variables $U_i \sim U(0,1)$ and normalize:

$$Z = \frac{\sum_{i=1}^{n} U_i - \frac{n}{2}}{\sqrt{n/12}} \tag{6}$$

The choice of n affects the accuracy of this approximation.

III. Testing Routines

We implement four tests to evaluate the accuracy and independence of the generated samples:

A. Distribution Tests

To assess how well the generated samples match the expected $N(0,1)$ distribution, we employ two tests:

1) Chi-Square Test: The Chi-square test compares the histogram of the generated samples with the expected probabilities from the standard normal distribution. The test statistic is computed as:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{7}$$

where O_i is the observed frequency in each bin and E_i is the expected frequency.

2) Kolmogorov-Smirnov (K-S) Test: The K-S test compares the cumulative distribution function of the sample data with that of the reference distribution.

B. Independence Tests

To verify the independence of the generated samples, we use two tests:

1) Autocorrelation Test: We compute the autocorrelation function (ACF). For independent samples, the ACF should be close to zero for all lags greater than zero:

$$R(k) = \frac{\sum_{t=1}^{N-k}(X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^{N}(X_t - \bar{X})^2} \tag{8}$$

where k is the lag and $\bar{X}$ is the sample mean.

2) Runs Test: The runs test checks for randomness in a binary sequence. The test statistic is:

$$Z = \frac{R - \bar{R}}{\sigma_R} \tag{9}$$

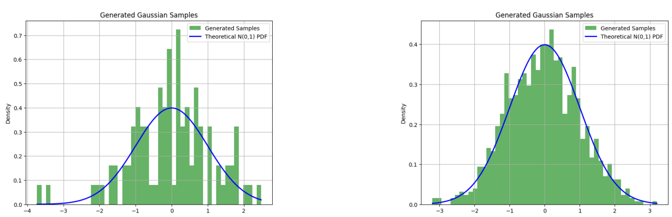
where R is the observed number of runs, $\bar{R}$ is the expected number of runs, and σ_R is the standard deviation of the number of runs.

IV. Results and Discussion

A. Comparison of Generation Methods

We compared the three methods for generating Gaussian random variables: For each method, we generated samples of size 100 and 1000, and compared the resulting distributions with the theoretical standard normal distribution.

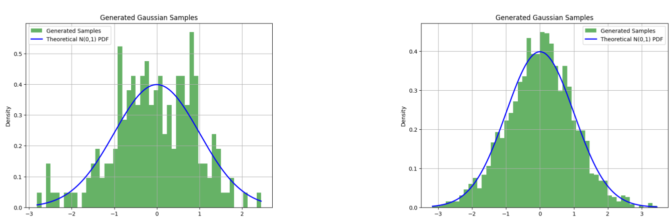
1) Newton's Method (Inverse-Transform Technique): Figure 1 shows the distribution of samples generated using Newton's method compared to the theoretical standard normal distribution for 100 and 1000 samples.



(a) 100 samples (b) 1000 samples

Fig. 1: Comparison of Newton's method generated samples with theoretical distribution

2) Box-Muller Transform: Figure 2 presents the distribution of samples generated using the Box-Muller transform compared to the theoretical standard normal distribution for 100 and 1000 samples.



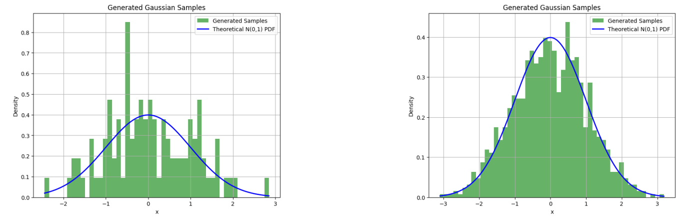
(a) 100 samples (b) 1000 samples

Fig. 2: Comparison of Box-Muller transform generated samples with theoretical distribution

3) Central Limit Theorem (CLT) Method: Figure 4 is the same illustration for CLT method

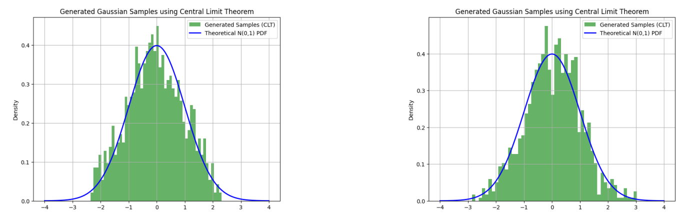
B. Effect of increasing Parameter N in CLT Method

As N increases, the sum or average of the random variables approximates a normal distribution more closely, resulting in reduced variance (if averaged) and a smoother, more bell-shaped distribution.



(a) 100 samples (b) 1000 samples

Fig. 3: Comparison of CLT method generated samples with theoretical distribution



(a) N=2 (b) N=12

Fig. 4: comparison of the effects of N

V. Distribution and Independence Analysis

- **Box-Muller Method:**
  - **K-S Test:** Test Statistic = 0.0234, p-value = 0.2215 (Normal distribution confirmed)
  - **Chi-Square Test:** Test Statistic = 14.5610, p-value = 0.1037 (Normal distribution fit)
  - **Runs Test:** Count = 1320, Expected = 1000.3677, z-statistic = 14.3034, p-value = 0.0000 (Independence questioned)

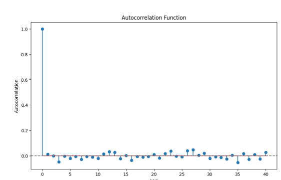


Fig. 5: Box-Muller method - Autocorrelation plot

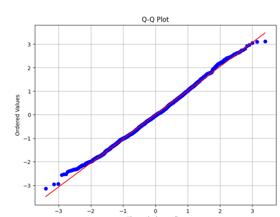


Fig. 6: Box-Muller method - Q-Q plot

- **Central Limit Theorem (CLT):**
  - **K-S Test:** Test Statistic = 0.0380, p-value = 0.1087 (Normal distribution confirmed)

- **Chi-Square Test:** Test Statistic = 10.7114, p-value = 0.2960 (Normal distribution fit)
- **Runs Test:** Count = 664, Expected = 500.3874, z-statistic = 10.3605, p-value = 0.0000 (Independence questioned)

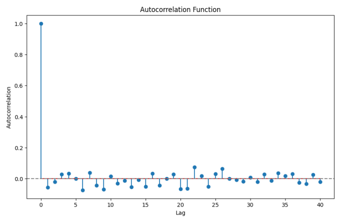


Fig. 7: CLT method - Autocorrelation plot

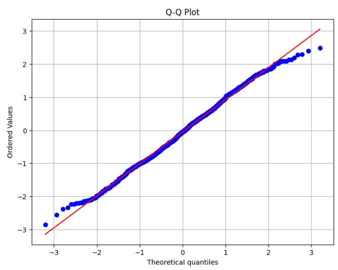


Fig. 8: CLT method - Q-Q plot

- **Newton's Method:**
  - **K-S Test:** Test Statistic = 0.0200, p-value = 0.8129 (Normal distribution confirmed)
  - **Chi-Square Test:** Test Statistic = 6.8410, p-value = 0.6537 (Normal distribution fit)
  - **Runs Test:** Count = 664, Expected = 500.3874, z-statistic = 10.3605, p-value = 0.0000 (Independence questioned)

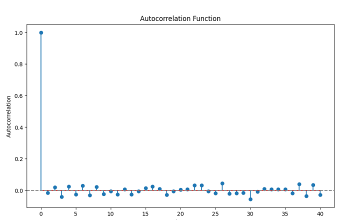


Fig. 9: Newton's method - Autocorrelation plot

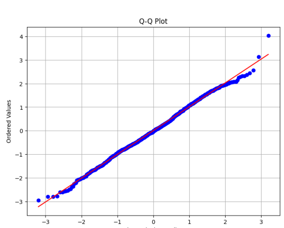


Fig. 10: Newton's method - Q-Q plot

VI. Conclusion

In this report, The results showed that while all three methods were capable of generating samples that approximate the standard normal distribution, the Box-Muller transform provided the better overall performance in terms of distribution accuracy and independence of samples.

Design and Implementation of a Multi-Server Queuing System for Supermarket Simulation

Erfan Bayat
S328390
erfan.bayat@studenti.polito.it

***Abstract*—This paper presents the design and implementation of a comprehensive supermarket simulation system using discrete event simulation. The system models multiple service sections including butchery, fresh food, general shopping areas, and checkout counters, with varying customer types and behaviors. The simulation incorporates dynamic queue management, customer patience modeling, and service time variations. Results demonstrate the system's ability to accurately model real-world supermarket dynamics and provide insights into optimal resource allocation and customer flow management.**

I. Introduction

This paper presents a discrete event simulation model that captures these complexities while maintaining computational efficiency. The simulation focuses on four key areas: butchery, fresh food, general shopping, and checkout services, with special attention to customer behavior patterns and service time variations.

II. System Model

A. Customer Types and Behavior

The system models four distinct customer types:

- Fresh Food Only (30% probability)
- Butchery Only (30% probability)
- Both Services (20% probability)
- Regular Shopping (20% probability)

Each customer type exhibits unique behavior patterns, service requirements, and patience thresholds. Service speeds are categorized as either **normal** (85%) or **slow** (15%), affecting service time distributions.

B. Queue Management

The system implements three distinct queue types:

$$Q = \{Q_{butchery}, Q_{fresh}, Q_{cashier}\} \quad (1)$$

Each queue has a finite capacity C, except for the general shopping area which implements an infinite server model. The queueing discipline follows these principles:

- First-Come-First-Served (FCFS) within each section
- Dynamic server allocation based on current load
- Patience-based retry mechanism for rejected customers

Queue capacity constraints are defined as:

$$|Q_i| \leq C_i \quad \forall i \in \{butchery, fresh, cashier\} \quad (2)$$

III. Implementation

A. Description of Simulator Structure

The simulator is designed with a Future Event Set (FES) that utilizes a priority queue implemented through Python's 'heapq' module. The FES organizes events based on their occurrence time and includes three primary event types:

- **Arrival**: Marks the time when a customer arrives at the system.
- **Section Entry**: Indicates the moment when a customer enters a service section.
- **Section Departure**: Signifies the time when a customer leaves a service section after completion.

Each event in the FES is formatted as a tuple:

$$\text{event} = (time, event\_type, event\_data)$$

B. Event Management

The system utilizes a priority queue-based Future Event Set (FES) for event scheduling:

Algorithm 1 Event Processing

```
while FES not empty do
    (time, event_type, data) ← HeapPop(FES)
    if event_type = arrival then
        HandleArrival(time, data)
    else if event_type = section_entry then
        HandleSectionEntry(time, data)
    else if event_type = section_departure then
        HandleSectionDeparture(time, data)
    end if
end while
```

C. Service Time Modeling

Basic service times follow exponential distributions with section-specific rates:

$$T_s = \begin{cases} \text{Exp}(\lambda_{\text{butchery}}) & \text{for butchery} \\ \text{Exp}(\lambda_{\text{fresh}}) & \text{for fresh food} \\ \text{Exp}(\lambda_{\text{other}}) & \text{for general shopping} \end{cases}$$

D. Cashier Time Calculation

The cashier service time is derived from other service times to reflect real-world dependencies. The total cashier time T_c is calculated as:

$$T_c = \max(1.0, (T_b + T_s) \cdot V) \tag{3}$$

where:

- T_b is the base time: $T_b = 0.2 \cdot \sum_{s \in S} T_s$
- T_s is the specialized item time: $T_s = 0.15(T_{\text{butchery}} + T_{\text{fresh}})$
- V is a random variation factor: $V \sim U(0.7, 1.3)$
- S is the set of all sections excluding cashier

The complete cashier service time formula can be expressed as:

$$T_c = \max\left(1.0, \left(0.2 \sum T_s + 0.15(T_b + T_f)\right) \cdot U(0.7, 1.3)\right) \tag{4}$$

This formulation ensures that:

- Cashier time scales appropriately with total shopping time
- Specialized items receive additional processing time
- Natural variations in the checkout process are captured
- Service time never falls below the minimum threshold

E. Customer Patience Model

The patience mechanism is modeled using a retry counter and maximum threshold (this criteria is only conditioned at cashier):

$$P(retry|attempts = n) = \begin{cases} 1 & \text{if } n < max\_retries \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

IV. A Customer's Journey Through the Supermarket: System Walkthrough

1) Customer Types and Initial Entry: Customers entering the supermarket are categorized into four types with their given shopping speed Upon entering, customers assess their needs.

2) Navigation Between Sections: Customers either join the queue, wait nearby, return later, or abandon fresh food items if the queue is full.

General Shopping Area: Operates without formal queuing, allowing free movement. Customers often use this area as a holding zone while waiting for specialty counters.

3) Customer Decision Points: **Queue Management**: Customers decide whether to join a queue, try again later, or abandon items based on wait times.

Service Priority: For multiple services, customers optimize their visits by considering sequence, time, and whether to abandon a service if it's too busy. this will ensure a dynamic approach from the customers and preventing unwanted traffic in the system

4) The Checkout Process: **Getting Ready for Checkout**: Customers ensure specialty counter visits and general shopping are complete before checking out.

Checkout Line Management: Customers select the shortest line or retry after 2–5 time units if lines are long. Repeated failures may result in abandonment of purchases.

Patience System: Customers attempt checkout lines up to three times before potentially leaving without purchase.

5) Special Considerations: **Customer Flexibility**: Realistic behaviors include adjusting plans based on queue lengths, returning to counters, and deciding based on time efficiency.

Time Management: Continuous evaluation of total store time, balancing wait times against shopping goals.

6) Journey Completion: **Successful Shopping Trip**: Involves obtaining all specialty and general items, checking out successfully, and spending a reasonable amount of time in-store.

Alternative Endings: May include partial completion, abandonment, modified plans, or extended shopping duration.

This system reflects real-world shopping dynamics, simulating customer navigation and decisions in response to service availability and time constraints.

V. Results and Analysis

The supermarket simulation was conducted with two different configurations to evaluate the performance under varying parameters.

A. Setup Comparison

Configuration 1: The first configuration involved initializing the `SupermarketSimulator` with:

- **Butchery Servers:** 2
- **Fresh Food Servers:** 2
- **Cashier Servers:** 3
- **Queue Capacity:** 5
- **Arrival Rate:** 0.8

The simulation ran for a total of 1000 time units, utilizing a specified random seed for reproducibility.

Configuration 2: The second configuration modified the parameters to:

- **Butchery Servers:** 1
- **Fresh Food Servers:** 2
- **Cashier Servers:** 2
- **Queue Capacity:** 7
- **Arrival Rate:** 0.9

This simulation also ran for a total of 1000 time units.

B. Customer Type and Section Performance Statistics

The tables below summarize performance metrics for various customer types and each section of the supermarket across both configurations. The first table presents key metrics for each customer type, and the second highlights section-specific performance, providing insights into both customer behavior and operational efficiency across different areas.

TABLE I
PERFORMANCE STATISTICS BY CUSTOMER TYPE

Customer Type	Count (1)	Average Time (1)	Count (2)	Average Time (2)
Butchery Customers	181	41.15 ± 21.06	73	70.71 ± 28.29
None Customers	127	23.48 ± 19.18	104	34.90 ± 19.07
Fresh Food Customers	185	32.56 ± 17.23	172	53.85 ± 21.55
Both (Butchery and Fresh Food) Customers	113	49.78 ± 23.58	57	86.60 ± 32.97

TABLE II
SECTION PERFORMANCE METRICS (CONFIGURATION 1)

Section	Arrivals (1)	Completions (1)	Rejections (1)	Avg. Service Time (1)	Avg. Waiting Time (1)
Butchery	380	373	49	4.78	7.59
Fresh Food	394	392	16	4.30	4.58
Cashier	614	606	112	4.59	5.48
Other	808	797	0	14.01	0.00

C. Efficiency Metrics

The efficiency of the supermarket operations can be summarized in Table IV for both configurations.

D. Simulation Test

The simulation was executed with the following parameters for Configuration 1:

- **Total Customers Served:** 814
- **Completed Customers:** 606
- **Rejected Customers:** 112
- **Active Customers:** 96
- **Average Time in System:** 36.43 ± 21.99 time units

For Configuration 2:

- **Total Customers Served:** 942
- **Completed Customers:** 406
- **Rejected Customers:** 153
- **Active Customers:** 383
- **Average Time in System:** 56.62 ± 29.54 time units

E. Plots of Simulation Results

The plots generated during the simulation are presented below:

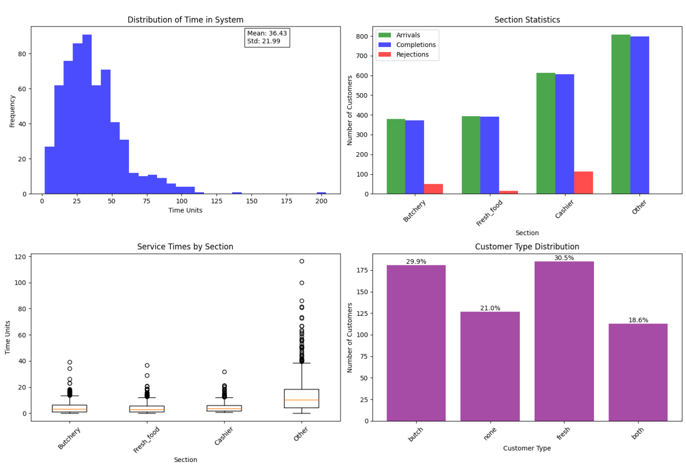


Fig. 1. Summary of the statistical observations in the simulation Configuration 1

TABLE III
SECTION PERFORMANCE METRICS (CONFIGURATION 2)

Section	Arrivals (2)	Completions (2)	Rejections (2)	Avg. Service Time (2)	Avg. Waiting Time (2)
Butchery	200	192	292	5.14	30.87
Fresh Food	415	408	51	4.70	10.83
Cashier	414	406	153	4.79	14.72
Other	940	926	0	14.65	0.00

TABLE IV
EFFICIENCY METRICS (CONFIGURATION 1)

Metric	Value
Completion Rate	74.4%
Rejection Rate	13.8%
Avg. Time in System	36.43 ± 21.99

F. Explanation of Confidence Interval Analysis Results

The analysis presents the performance metrics for a system, focusing on various aspects such as time in the system, completion rate, rejection rate, service time, waiting time, and utilization across different sections (butchery, fresh food, cashier, and other services). The confidence intervals were calculated using the t-distribution, and the simulation was run 100 times with different random seeds (for this part of the report we only focus on Configuration 1).

1) : Overall Metrics

- **Time In System:** The average time a customer spends in the system is approximately 28.72 units. The confidence interval, ranging from 28.316 to 29.129, indicates that we can be 95% confident that the true mean time in the system lies within this range. The standard error of 0.205 suggests a relatively low variability in the estimates.
- **Completion Rate:** The average completion rate is about 0.967, which means that a high percentage of customers complete their transactions successfully. The confidence interval (0.965 to 0.969) shows that we can be 95% confident this true mean lies within these bounds, with a very small standard error of 0.001.
- **Rejection Rate:** The average rejection rate is very low at 0.002, suggesting that almost all customers are accepted. The confidence interval (0.001 to 0.003) further confirms the reliability of this metric, with a standard error of 0.000.

2) : Section-Specific Metrics

Butchery:

- **Average Service Time:** Customers in the butchery section have an average service time of approximately 4.988 units, with a confidence interval between 4.929 and 5.046, indicating consistent service speed.
- **Average Waiting Time:** The average waiting time is about 2.847 units, with a confidence interval of 2.640 to 3.054, indicating that customers can expect to wait within this time range.
- **Utilization:** The butchery section is utilized at about 62.9%, with a confidence interval of 61.8% to 64.1%, reflecting efficient use of resources.

TABLE V
EFFICIENCY METRICS (CONFIGURATION 2)

Metric	Value
Completion Rate	43.1%
Rejection Rate	16.2%
Avg. Time in System	56.62

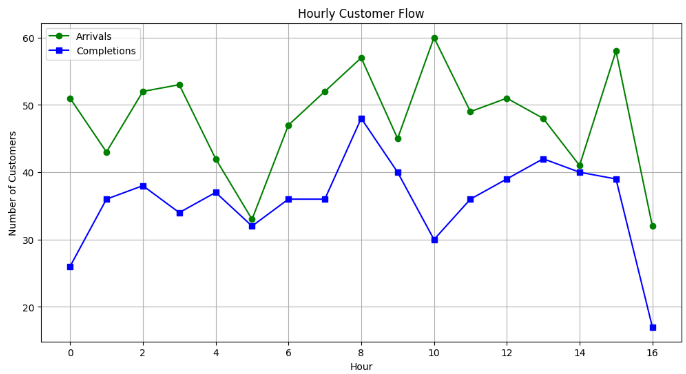


Fig. 2. Hourly customer flow chart for Configuration 1

Fresh Food:

- **Average Service Time:** The average service time for fresh food is around 3.988 units, with a confidence interval from 3.938 to 4.038, indicating quick service.
- **Average Waiting Time:** Customers typically wait about 1.405 units of time, with the confidence interval ranging from 1.301 to 1.509.
- **Utilization:** The utilization rate is 50.5%, suggesting that the section operates at half of its capacity.

Cashier:

- **Average Service Time:** The cashier's average service time is approximately 4.559 units, with a confidence interval of 4.525 to 4.593, showing reliable service speed.
- **Average Waiting Time:** The average waiting time in this section is about 2.829 units, with a confidence interval from 2.627 to 3.031.
- **Utilization:** The utilization is relatively high at 75.3%, with a confidence interval of 74.4% to 76.2%, indicating effective use of cashier resources.

Other Services:

- **Average Service Time:** The average service time is significantly higher at 14.657 units, with a confidence interval of 14.512 to 14.803.
- **Average Waiting Time:** Notably, the average waiting time is 0.000, indicating that there are no delays in this section, possibly due to its nature.
- **Utilization:** The utilization here is at full capacity (1.000), suggesting that resources are being maximally employed.

3) Summary: The results provide a comprehensive overview of the system's efficiency and customer experience across different service sections. The confidence intervals, calculated using the t-distribution, confirm the reliability of the averages, suggesting that customers can expect consistent service times and high completion rates, while also indicating areas of strong performance, such as the cashier and butchery sections. The use of 100 different seeds in the simulation enhances the robustness of these findings.

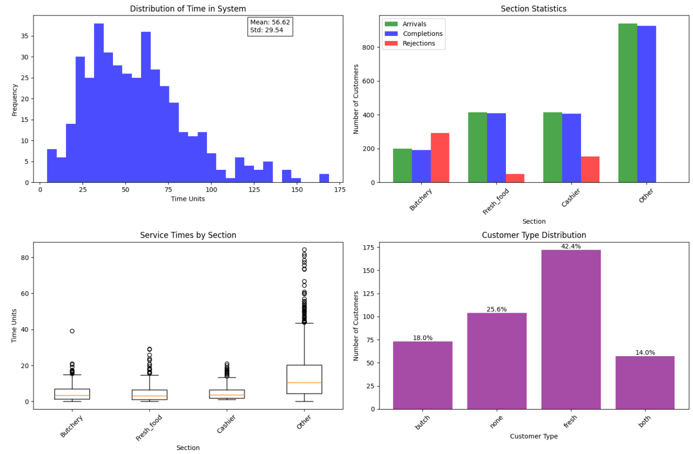


Fig. 3. Summary of the statistical observations in the simulation Configuration 2

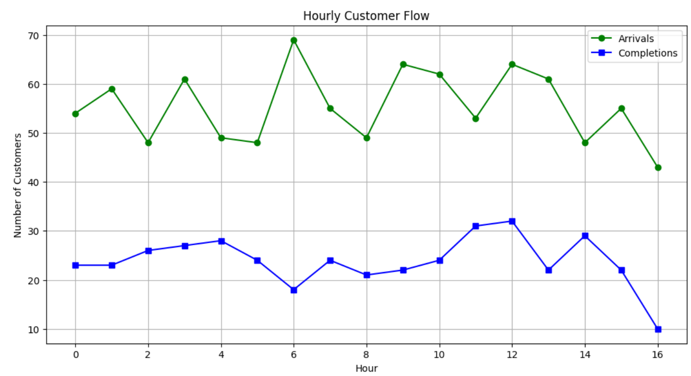


Fig. 4. Hourly customer flow chart for Configuration 2

VI. CONCLUSION

Conclusion In this study, we conducted a simulation of supermarket operations to evaluate the impact of different configurations on customer flow, service efficiency, and overall performance. By comparing two distinct configurations, we identified key metrics such as completion and rejection rates, average service times, and customer satisfaction across various sections of the supermarket.

The findings indicate that Configuration 1, which employed a higher number of servers and a lower arrival rate, achieved superior performance with a higher completion rate and lower average time in the system. In contrast, Configuration 2, while showing some resilience in handling customer types, faced increased rejection rates and longer average service times, highlighting the trade-offs between service capacity and customer demand.