Simulating my Local Coffee Shop

Dec 5, 2024

Aaron Tsui

Confidential

Copyright ©

Problem Statement

- ***What possible factors can be manipulated in order to :***
  - ***Reduce lost customers from balking/reneging?***
    - Minimizing balking/reneging can allow management to allocate funds for employee bonuses/upward mobility, which can lower overall turnover
  - ***Limit employee turnover due to overwork, stressful conditions, etc?***
    - The turnover rates in the restaurant and hospitality industries are significantly higher than other industries like computing or technology
    - High turnover rate increases hiring and training costs, which can both reduce employee productivity and cost the business money through job ads, the time spent during interviews, etc.
    - Employees leaving in peak hours can bottleneck potential sales and/or lead to more customers balking/reneging

Confidential

Copyright ©

Simulation Methodology

Time Management

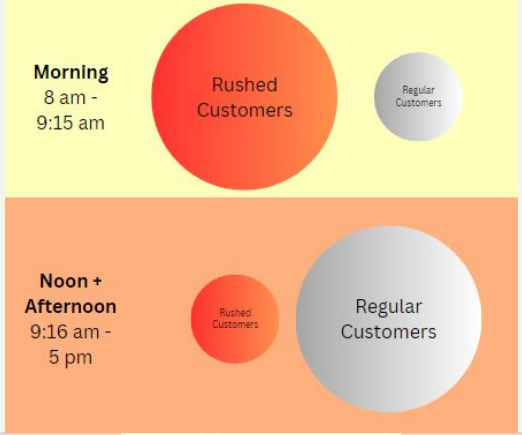
- Simulation is split between 2 phases of high traffic time periods (mornings) and low traffic time periods (other times)
- Total simulation run time is set to 540 minutes or 9 hours, representing the actual hours my local coffee shop is open in a single day

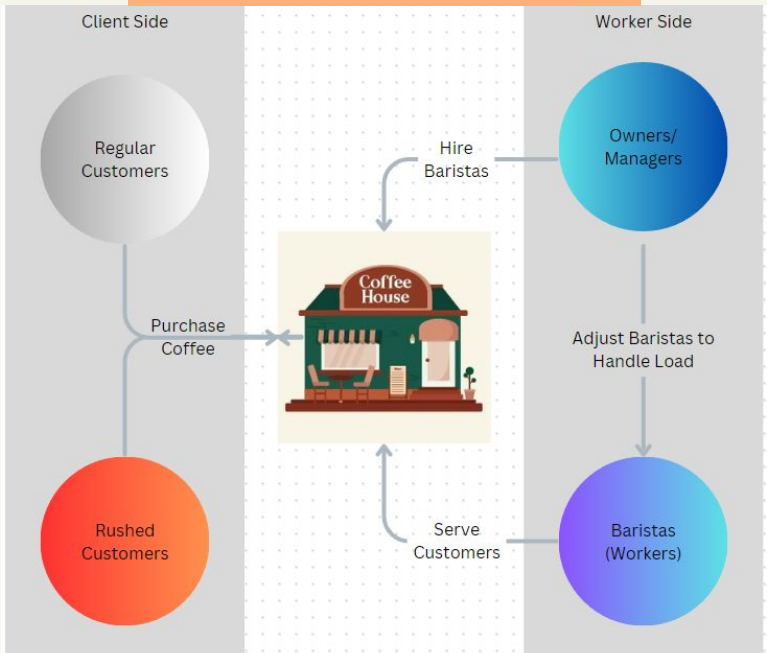
Performance Metrics

- Key metrics include customer wait times, customers served per barista, baristas on reserve (backup), frequency of barista count changes/modifications

Methodology

- Using Python and Discrete-event simulation

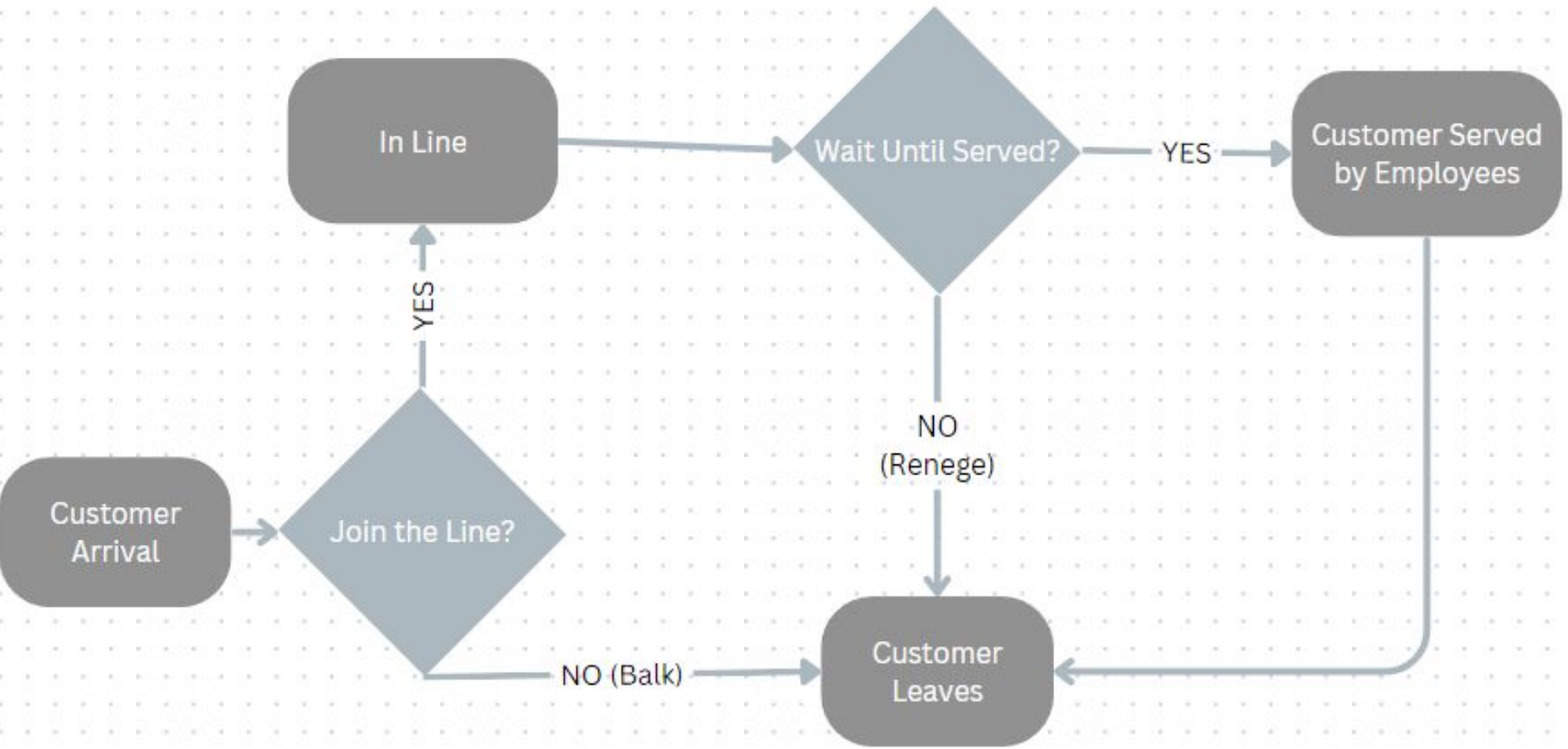




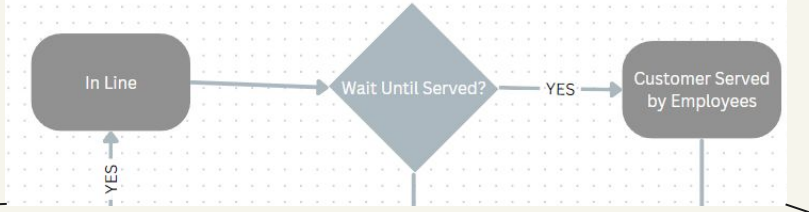
Confidential

Copyright ©

Process Flow



A Closer Look



Whenever another customer joins the queue, customers currently in the queue can renege.

In Line

YES

Wait Until Served?

Renege

Customer Leaves

Yes

Customer Served by Employees

Customers only renege if rushed. Renege chance is lower if they are

- halfway through the queue to the beginning of the queue
- the queue is short

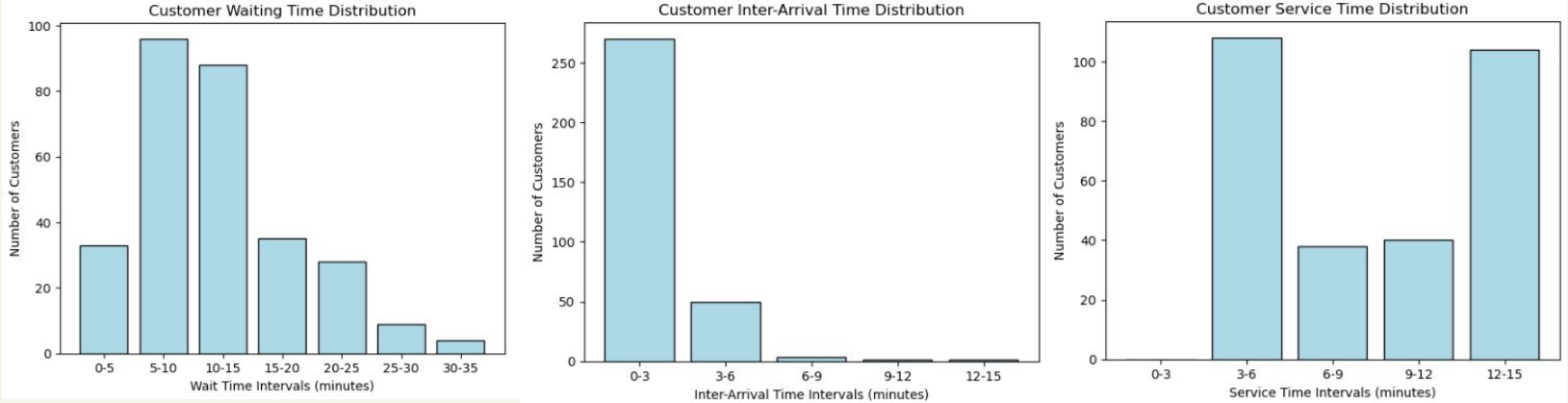
Code Review

```python
# Parameters
baristas = 3
simulation_hours = 9
obtain_reproducible_results = True

morning_rush = 1.25 # hours (how long the morning rush lasts)
prushed = 80 # % of customers that are rushed in the morning

prushed_a = 30 # % of customers that are rushed at other times of the day

# Adjust Barista Thresholds
# Add:     Current Queue > 8 & Current Baristas < 5
# Remove:  Current Queue < 3 & Current Baristas > 2
mod_baristas_q_threshold = [3, 8]
mod_baristas_b_threshold = [2, 5]

minimum_service_time = 5
mean_service_time =  12
maximum_service_time = 15
mean_inter_arrival_time = 2
```

```python
highest_length_cust = None
ran = 0

if caseid_queue.qsize() > 2:
    reduced_qmax = prev_cust[2] // 2
else:
    reduced_qmax = prev_cust[2]

for position in range(caseid_queue.qsize()):
    if previous_casetype == 'rushed' and position != 0 and reduced_qsize > 0:
        latest_rushed = position
        highest_length_cust = prev_cust
        reduced_qsize -= 1
        ran = random.random()

if highest_length_cust and caseid_queue.qsize() > reduced_qmax and ran > 0.5:
    print(f"😓Customer #{caseid-1} Reneges | Time:{env.now} | {caseid_queue.qsize()} in line")
    ren += 1
    env.process(event_log_append(env, caseid-1, env.now, 'renege', event_log))
    ren_flag = True
else:
    ren_flag = False
```

```python
def adjust_baristas(env, caseid_queue, dynamic_resource, barista_allocation_interval=10):
    while True:
        yield env.timeout(barista_allocation_interval)
        current_queue_length = caseid_queue.qsize()
        new_capacity = dynamic_resource.capacity

        if current_queue_length > mod_baristas_q_threshold[1] and dynamic_resource.capacity < mod_baristas_b_threshold[1]:  # Increase baristas
            new_capacity += 1
            print(f"                                          ➕Baristas")
        elif current_queue_length < mod_baristas_q_threshold[0] and dynamic_resource.capacity > mod_baristas_b_threshold[0]:  # Decrease baristas
            new_capacity -= 1
            print(f"                                          ➖Baristas")

        dynamic_resource.adjust_capacity(new_capacity)
```

Insights from Results



- Suggests min. 2-3 baristas should be on reserve (backup) for efficient serving of customers on half-shifts of 4 hours as to meet demand and minimize customer loss
  - Suggested Extra Barista Shifts:
    - Shift 1: 8am - 12:45pm (4h45m)
    - Shift 2: 1:15pm - 5pm (4h45m)
- Simulated Barista Modifications: 6 sets of additions and removals
  - Morning covers 1 long and 1 short high traffic period
  - Noon + Afternoon covers 4 short high traffic periods
- Line is moderately sized at closing so customer wait times near closing may be artificially inflated due to lower number of baristas serving customers

Confidential

Copyright ©

Thank you for listening!

Confidential

Copyright ©