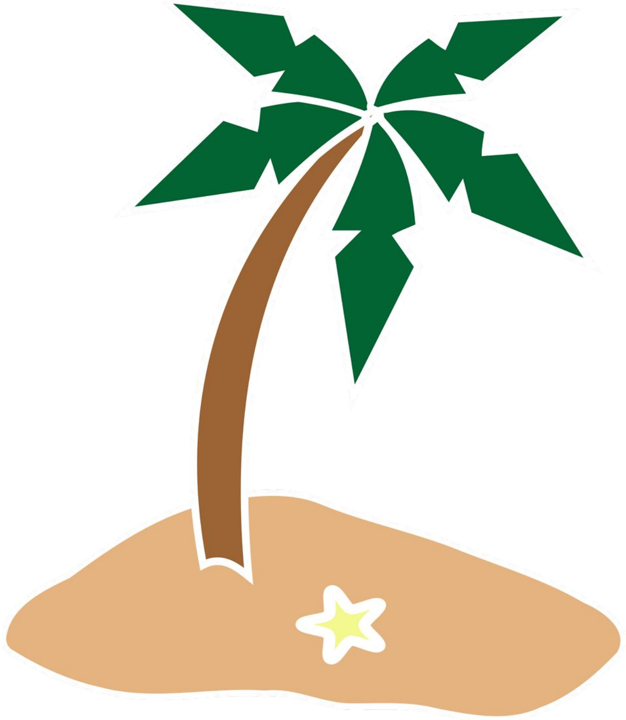
Coffee Shop Simulation

Griffin Arnone, Anhua Cheng, Bailey Scoville
MSDS 460 | Winter 2024

About the Coffee Shop

- Objective: Maximize profits
- Concerns:
  - Labor Cost - no tips, $18/hr
  - Balking - lost revenue
  - Reneging - lost revenue
  - Tourism-heavy location
    - Working professionals
    - Vacationer and retirees



Customer Profiles

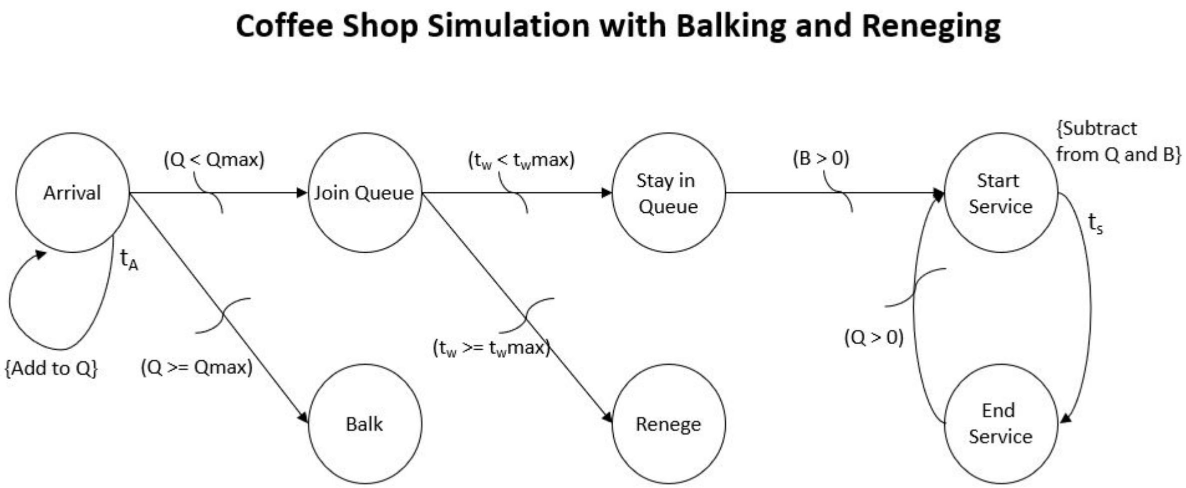
70% - Not in rush, low reneging risk

Retirees, Vacationers

30% - In a rush, HIGH reneging risk

Working professionals

Event Graph



Q = number of customers in queue
B = number of baristas available
t_A = time until next customer arrives
t_s = time to serve customer
t_w = time in line
Use random number generators to determine t_A and t_s for each event
Balking parameter Qmax shows the longest line a customer will join
$t_w$max shows the longest a customer would wait in line before leaving

Customer Behaviors

Morning Shift
- Arrival: 1 min
- Drink + Food
- Balking: > 6 customers
- Reneging: 30% < 6 min

Afternoon Shift
- Arrival: 5 min
- Drink Only
- Balking: > 10 customers
- Reneging: 30% < 10 min

Evening Shift
- Arrival: 7 min
- Drink Only
- Balking: > 10 customers
- Reneging: 30% < 10 min

Code Review

Balking Condition

```
#create function for random arrivals and balking condition

def arrival(env, caseid, caseid_queue, event_log):
    caseid = 0
    while True:
        inter_arrival_time = round(60*np.random.exponential(scale = mean_inter_arrival_time))
        print("Next arrival time: ", env.now + inter_arrival_time)
        yield env.timeout(inter_arrival_time)
        caseid += 1
        time = env.now
        activity = 'arrival'
        env.process(event_log_append(env, caseid, time, activity, event_log))
        yield env.timeout(0)
        #define balking condition
        if caseid_queue.qsize() < balk_queue_length:
            caseid_queue.put(caseid)
            print("Customer joins queue --> caseid =",caseid,', time = ',env.now,', queue_length =',caseid_queue.qsize())
            time = env.now
            activity = 'join_queue'
            env.process(event_log_append(env, caseid, time, activity, event_log))
            env.process(service_process(env,  caseid_queue, event_log))
        else:
            print("Customer balks --> caseid =",caseid,', time = ',env.now,', queue_length =',caseid_queue.qsize())
            env.process(event_log_append(env, caseid, env.now, 'balk', event_log))
```

Code Review

Reneging Condition & Barista Service Adjustment

```python
#create function for flow of service and renege condition and number of baristas

def service_process(env, caseid_queue, event_log):
    with baristas_on_shift.request() as req:
        yield req
        if not caseid_queue.empty():
            queue_length_on_entering_service = caseid_queue.qsize()
            caseid = caseid_queue.get()
            wait_time = env.now - queue_length_on_entering_service * (mean_service_time * 60)
            if wait_time > (max_wait_time * 60) and random.random() <= 0.3:    #define reneging condition
                print("Customer", caseid, 'left the queue after waiting for', wait_time, 'minutes')
                env.process(event_log_append(env, caseid, env.now, 'renege', event_log))
            else:
                adjusted_mean_service_time = mean_service_time / baristas #adjusts service time based on baristas working
                adjusted_max_service_time = max_service_time / baristas #adjusts service time based on baristas working
                print("Begin_service --> caseid =",caseid,', time = ',env.now,', queue_length =', queue_length_on_entering_service)
                env.process(event_log_append(env, caseid, env.now, 'begin_service', event_log))
                service_time = round(60*random_service_time(min_service_time, mean_service_time, max_service_time))
                yield env.timeout(service_time)
                queue_length_on_leaving_service = caseid_queue.qsize()
                print("End_service --> caseid =",caseid,', time = ',env.now,', queue_length =', queue_length_on_leaving_service)
                env.process(event_log_append(env, caseid, env.now, 'end_service', event_log))
```

Applications to Management

Morning Shift			
Assumptions			
Average Revenue per Customer (drink + food)	10	10	10
Baristas Hourly Wage	18	18	18
Variables			
# of Baristas Working	1	2	3
Mean Interval Arrival Time (min)	1	1	1
Min Service Time (min)	1	1	1
Mean Service Time (min)	4	4	4
Max Service Time (min)	8	8	8
Balking Queue Length	6	6	6
Max Wait Time (min)	6	6	6
Simulation Outputs			
# of Customers Arrived	232	232	216
# of Customers Serviced	58	115	155
Min Wait Time (min)	0	0.02	0
Mean Wait Time (min)	15.46	4.96	2.62
Max Wait Time (min)	29.73	13.45	9.88
# of Customer Balked	146	56	15
# of Customer Left the Queue	21	54	44
Estimated Gross Revenue	580	1150	1550
Estimated Cost	72	144	216
Estimated Income	**508**	**1006**	**1334**
Estimated Lost Revenue	1670	1100	590

Observations & Recommendation:

- To maximize profit, 3 baristas is the optimal hiring level given the high customer traffic flow in the morning
- With 3 baristas, there is an estimated revenue loss of $590 due to customers that did not wish to wait in line or left the queue after waiting for too long. That accounts for almost 40% of the gross revenue
- In addition to hiring 3 baristas, we recommend implementing measures such as offering pre-order option to accommodate time-sensitive customers and minimize lost business

Applications to Management

Afternoon Shift			
Assumptions			
Aerage Revenue per Customer (drink)	5	5	5
Baristas Hourly Wage	18	18	18
Variables			
# of Baristas Working	1	2	3
Mean Interval Arrival Time (min)	5	5	5
Min Service Time (min)	1	1	1
Mean Service Time (min)	2	2	2
Max Service Time (min)	5	5	5
Balking Queue Length	10	10	10
Max Wait Time (min)	10	10	10
Simulation Outputs			
# of Customers Arrived	38	45	41
# of Customers Serviced	23	31	28
Min Wait Time	0	0	0
Mean Wait Time	6.27	0.03	0
Max Wait Time	39.67	0.9	0
# of Customer Balked	0	0	0
# of Customer Left the Queue	13	14	13
Estimated Gross Revenue	115	155	140
Estimated Cost	72	144	216
Estimated Income	**43**	**11**	**-76**
Estimated Lost Revenue	65	70	65

Observations & Recommendation:

- To maximize profit, 1 baristas is the optimal hiring level given the medium customer traffic flow in the afternoon
- With 1 baristas working, the average wait time is 6.27 minutes while the maximum wait time is almost 40 minutes (39.67), which seems quite long
- Validate population profile in the neighborhood to confirm if majority of the potential customers are not time sensitive in the afternoon
- In addition to hiring 1 baristas, the coffee shop can keep customers updated on their orders while they wait. Or they can diversify product offerings and sell products those customers might be interested in shopping while they wait

Applications to Management

Evening Shift			
Assumptions			
Aerage Revenue per Customer (drink)	5	5	5
Baristas Hourly Wage	18	18	18
Variables			
# of Baristas Working	1	2	3
Mean Interval Arrival Time (min)	7	7	7
Min Service Time (min)	1	1	1
Mean Service Time (min)	2	2	2
Max Service Time (min)	5	5	5
Balking Queue Length	10	10	10
Max Wait Time (min)	10	10	10
Simulation Outputs			
# of Customers Arrived	26	32	27
# of Customers Serviced	17	24	19
Min Wait Time	0	0	0
Mean Wait Time	0.25	0	0
Max Wait Time	3.97	0	0
# of Customer Balked	0	0	0
# of Customer Left the Queue	9	8	7
Estimated Gross Revenue	85	120	95
Estimated Cost	72	144	216
Estimated Income	**13**	**-24**	**-121**
Estimated Lost Revenue	45	40	35

Observations & Recommendation:

- To maximize profit, 1 baristas is the optimal hiring level given the low customer traffic flow in the evening
- With 1 baristas working, the net profit is only $13 per day
- Per our sensitivity analysis, the coffee shop will experience net loss if the average customer arrival interval time increases to 11 minutes
- More research is needed to validate the customer traffic flow during this shift before committing the resources

Thank you

References

BBC. 2017. “British Queuing and 'The Power of Six.'” BBC. https://www.bbc.com/news/uk-38990535.

Dagkakis, G., and C. Heavey. 2017. “A Review of Open Source Discrete Event Simulation Software for Operations Research.” *Journal of Simulation* 10, no. 3 (December): 193-206. https://doi.org/10.1057/jos.2015.9.

Helfand, Zach. 2023. “Has Gratuity Culture Reached a Tipping Point.” *The New Yorker*, December 25, 2023. https://www.newyorker.com/magazine/2023/01/01/has-gratuity-culture-reached-a-tipping-point.

Peyman, Mohammad, Pedro Copado, Javier Panadero, Angel A. Juan, and Mohammad Dehghanimohammadabadi. 2021. “A Tutorial on How to Connect Python with Different Simulation Software to Develop Rich Simheuristics,” Presentation at the 2021 Winter Simulation Conference (WSC), Phoenix, AZ. https://doi.org/10.1109/WSC52266.2021.9715511.