Modelling & Simulation Restaurant

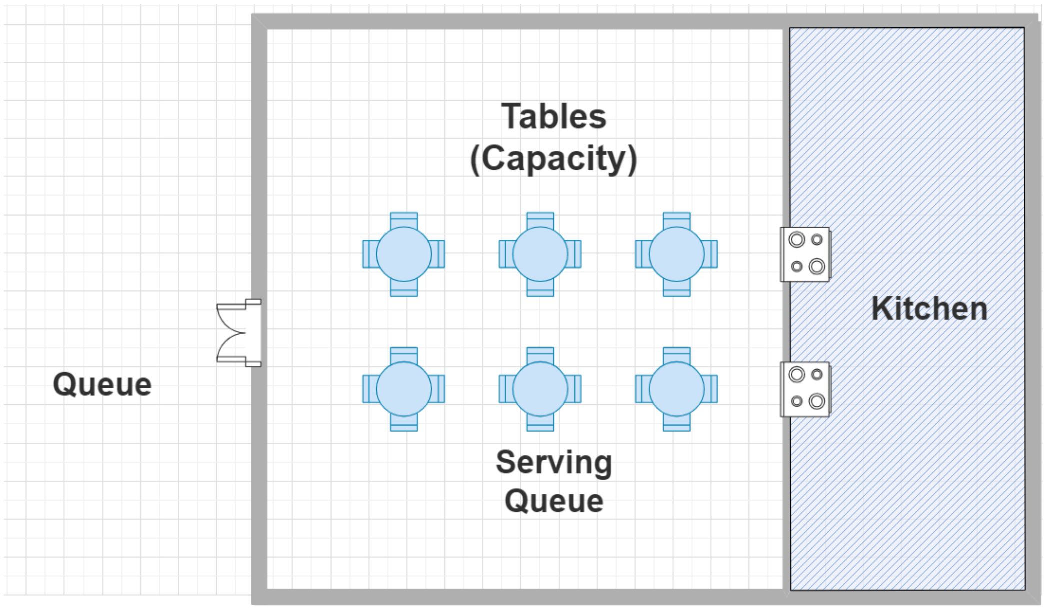
Ahmed Hakan Demirtaş
202010020113

ABSTRACT

This is a simulation project report for a restaurant in python. We will use simpy, random and panda libraries for our simulation. This Simulation will contain a day of a restaurant where customers come wait in a queue and join. Order food and wait for their food to delivered. After they will be served and have meal time as well. Finally after eating they will leave. This will give us the time taken to serve one customer.

INTRODUCTION

First let’s look at our structure,



We have a restaurant with a capacity of 6 tables and 2 kitchen. Outside we will have a queue if restaurant is full where each customer have to wait. In our kitchen we will have 2 chefs and 2 waiters, which will bring and serv the customers when needed.

The time each customer spends in the restaurant will be recorded.. This will give us the time each customer has spent in the restaurant.

The purpose of this simulation is to analyze the changes in customer waiting times when we increase either the kitchen capacity or the number of waiters.

After simulating it multiple times we will analyze the results to determine which capacity increase is more effective.

DISCUSSION

The main discussion will be the time changes when we change the capacity of kitchen or waiters.

Which one is more effective than the other ?

Which has less waiting time on each customer ?

In this report we will build our simulation and see the results with changing times on each increase.

CODE STRUCTURE

A. *Implement Libraries*

We said that we will use different libraries in our project.

Lets begin with simpy, Why do we need simpy ?

Simpy is a powerful Python library for process-based simulation, there for it will make different progress.

```
import simpy as sp
import random as rd
import pandas as pd
```

B. *Making the restaurant*

Here we will create our Restaurant class and add each feature to it. First we will add the capacities for tables, kitchen, waiters, … .

```
class Restaurant(object):
    def __init__(self, env, num_tables, table_capacity,
kitchen_capacity, num_waiters, menu_data):
        self.env = env
        self.tables = sp.Resource(env, capacity=num_tables)
        self.kitchen = sp.Resource(env, capacity=kitchen_capacity)
        self.waiters = sp.Resource(env, capacity=num_waiters)
        self.food_container = sp.Container(env, init=100, capacity=200)
        self.queue = sp.Store(env, capacity=MAX_QUEUE_SIZE)
        self.menu = sp.Store(env)
        self.initialize_menu(menu_data)
```

As we see we also use Store for having our queue outside the restaurant.

Init method for restaurant initializates resources, containers, and stores for tables, kitchen, waiters, food, queue, and menu items within the restaurant simulation.

C. *Customer Class*

With the Customer class, we will have customer visiting Restaurant, waiting at queue, ordering food, waiting for food, eating time and leaving. Below code is the init method for Customer class, Its includes initialization of the environment for the customer. Gets the unique number for customers.

```
class Customer(object):
    def __init__(self, env, num, restaurant):
        self.env = env
        self.num = num
        self.restaurant = restaurant
        self.result_time = 0
```

D. Making a Order

When the customers arrive at the restaurant they will given a unique number. This number will identify them when ordering.

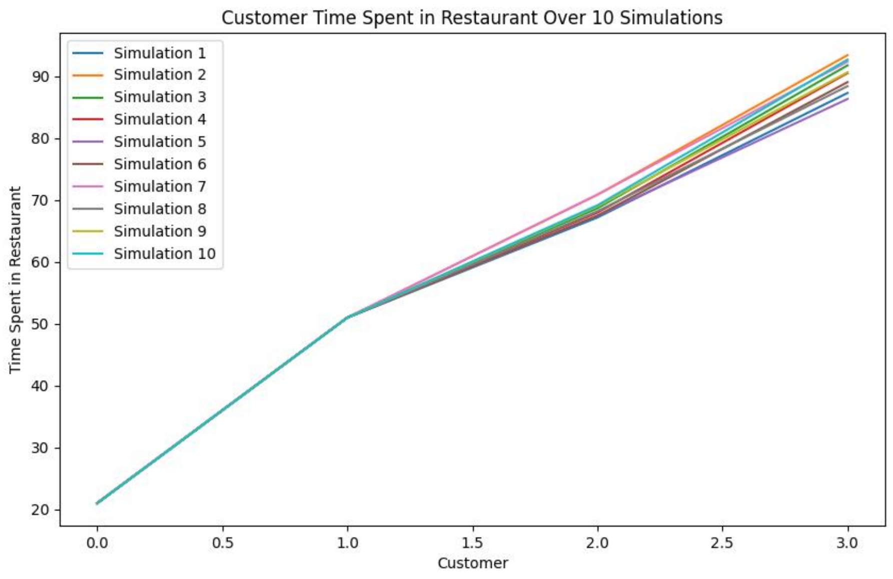
```
request = self.restaurant.tables.request()
        yield request
        arrival_time = self.env.now
        print("Customer", self.num, "has been seated at time",
round(self.env.now, 1))
```

Results of simulations

We have created our classes and made the simulation.

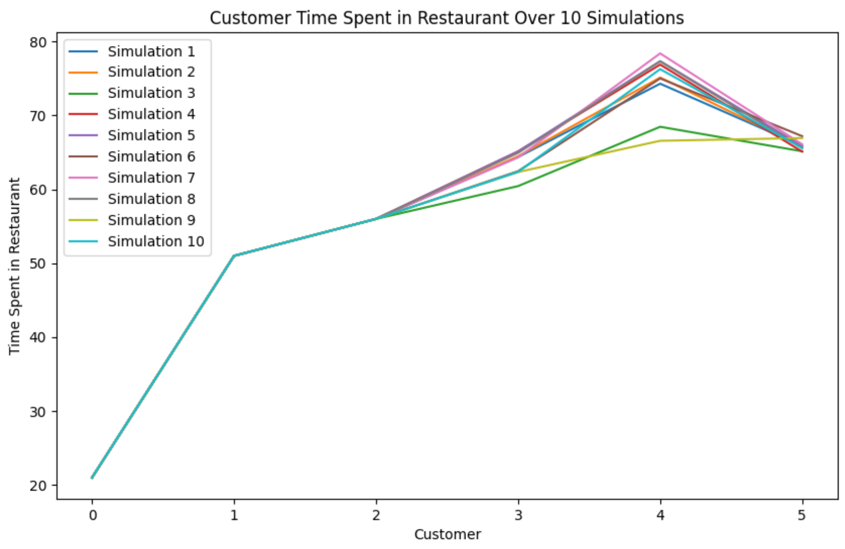
By default we have this result after simulating the project multiple times.

We see that the graph is linear and doesn't have a specific break point.



A. One increase of Kitchen Capacity

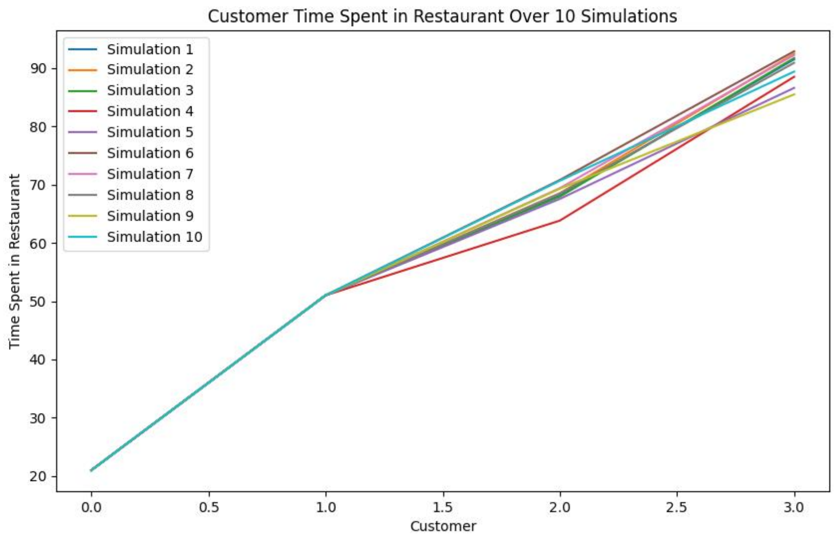
This simulation result are the results where we increased the kitchen capacity by one.



The results are showing that each customers time spent in the restaurant increases after a break point where the kitchen start cooking and every customer who joins after will have to wait more than the first customers.

B. Increasing Table Capacity

When we increase Table capacity with 2 more, the result graph is similar to the standard simulation result. This will also effect the timing of each customer which will be served.



SOLUTION

For the solution of our simulation project, we changed each capacity and simulated the result as graphs. The main object is that every simulation had his own linear or break down points in his graph. We can say that increasing kitchen capacity first decreases the time of the customers and has his prime time of average. How ever after some time we see that it creates a break point and decreases again which will effect the average of the customer waiting at the restaurant. The table capacity graph shows us that it has a average good time of serving the customers and doesn't have ups and downs. This shows us that it doesn't have a breakdown point and its linear.

CONCLUSION

As Conclusion we saw each step to build our simulation of our restaurant and tried to simulate which capacity is much more effective to our average customer time spent in the restaurant. The capacity of kitchen seemed to improve the time very good but it wasn't constant. Where the table capacity simulation result graph kept constant and resulted better to the average time of customer. Therefore I can say that increasing the capacity of tables is much more effective than increasing the kitchen size.

APPENDIX

The Code for the full simulation of our Restaurant is down below.

REFERENCES

HTTPS://WWW.KAGGLE.COM/DATASETS/SUKHMANDEEPSINGHBRAR/INDIAN-FOOD-DATASET

HTTPS://HARRYMUNRO.HASHNODE.DEV/SIMULATING-A-HOSPITAL-SERVING-100000-PATIENTS-IN-100-LINES-OF-PYTHON