Term Project: Restaurant Process Simulation

Chicago 1 Group:
Humberto Holanda, Nikita Sharma, and Sergio Valentini

MSDS 460-DL : Decision Analytics
Northwestern University
June 02, 2024

Abstract and Problem Definition

Most business processes like restaurant running can be described as separate discrete events. This study aims to develop a discrete event simulation tool for restaurant owners to help them better understand their business processes and make data-driven decisions. The discrete-event simulation of this study focuses on a system's processes at a moderate abstraction level. The simulation, built using the SimPy library, represents a typical day of restaurant operations, including customer arrival, food ordering, dining, bill payment, and resource utilization. The goal is to maximize profits, optimize staffing, and explore expansion opportunities. Using Monte Carlo discrete event simulations, we aim to determine the optimal number of staff and the potential for restaurant expansion based on various simulation parameters and processes. The results will provide key performance indicators (KPIs) such as total customers served, customers who left due to impatience, and financial statistics.

Introduction

The restaurant industry is characterized by variability and uncertainty in customer demand, service times, and operational processes, though it is not historically known as a data-focused industry. However, the rapid proliferation of data science tools is bringing large-scale changes to old-school business models. As noted by Debijt, Spiliotopoulou, and De Vries (2023), to maintain competitiveness, restaurants must adapt their approach to business decisions by redirecting their focus from food costs only to broader revenue management, capitalizing on opportunities for expansion and growth.

Simulation tools, like discrete event simulations, can empower restaurant managers to simulate operational scenarios and assess the influence of specific resources on overall restaurant performance without disrupting real-world operations. As part of the revenue management toolkit, these tools enable the use of factual data to determine the impact on profit and decision-making by measuring objective metrics such as customers served per minute and the effect of queues and service levels on the business's profitability.

Literature review

Discrete event simulation using SimPy is a popular research topic. Restaurant process simulations are less common, though there is existing research that the team found relevant to this study. Zinoviev (2024) published an excellent (and humorous) tutorial on utilizing the SimPy library to simulate Dijkstra's classic dining philosophers problem, with forks as SimPy Resources and the various philosopher states SimPy Processes. Zhané (2024) demonstrated a SimPy movie theater discrete event simulation with a restaurant-like process flow including purchase_ticket(), check_ticket(), and sell_food(), each of which requests different resources.

Research Design, Algorithms, Modeling Methods, Implementation and Programming

This research employed Monte Carlo simulations to optimize restaurant staffing (number of cooks and servers) and maximize profits. It also investigated expansion opportunities based on the restaurant's capacity to serve customers (number of tables). The SimPy Python library was the primary tool for creating this discrete event simulation, offering functions to build agents, processes, and resources, and a simulation environment for their interaction. The Pandas library was used to create a data frame for analysis, facilitating a comprehensive understanding of the simulation results. On the other hand, the NumPy library was instrumental in generating random number distributions for specific simulation parameters. The Queue library established a First-in, First-out data structure, and Functools aided in event log creation.

Several agents, resources, and processes are critical in the restaurant process model. Customers are the primary agents in this simulation. They are created using the user-defined **Customer** class and infinitely generated by the **arrival_waiting** generator function for the length of the simulation. This function also determines if a customer in the queue will be a dine-in or takeout customer based on the DINE_IN_PROB and TAKEOUT_PROB probability simulation parameters.

In addition, the functions **dinein_service_process** and **takeout_service_process** define and yield the activities in each service queue. Both functions request the appropriate required resources and establish a random patience factor for each customer to simulate the varying degrees to which customers will wait to be served. Simulation resources include tables, servers, and cooks, each of which is requested by dine-in or takeout customers.

A dictionary of simulation parameters is established to quickly test what-if scenarios, which is included in the env.process call to begin the simulation. The functions trace and event_log_append create a parseable event log, a helpful tool for simulation analysis. The simulation start is triggered with **run_multiple_simulations**, which will run the simulation 100 times and store the results in **restaurant_simulation.csv**. Each run provides a different outcome due to the random nature of the inputs, which helps in understanding the distribution of possible outcomes.

After the simulation ends, two Pandas data frames are created for analysis: **totals_df**, which stores the totals of each activity type for each simulation as well as revenue, cost, and profit figures, and **event_log_df_all**, which contains the complete event log of all simulation runs.

Results and Recommendations

The experiments considered the effect on profit as the number of employees (servers and cooks) and critical resources (tables) varied. We calculated the mean profit for several scenarios using Monte Carlo analysis, as shown in Table 01 below. All other parameters remained constant.

The results show that the optimal solution for the simulation is to maintain the status quo, i.e. use 1 server, 2 cooks, and 10 tables. The small increase in revenue generated by adding additional tables does not offset the increased costs of hiring additional staff to serve and cook for more customers. Therefore, the recommendation is hold off on expansion at this time. It is interesting to notice that the solution for this problem cannot be solved by the classic queueing theory, given the complexity of the formulation.

Table 01. Experiment design for the restaurant simulation and the effect on the restaurant profitability (average and standard deviation).

Experiment	Servers	Cooks	Tables	Revenue avg. ($)	Cost avg. ($)	Profit avg ($)	Profit std dev ($)
01	01	02	10	$903.10	$360	**$543.10**	$133.50
02	03	04	15	$1001.20	$840	$161.20	$148.83
03	02	03	12	$991.60	$600	$391.60	$144.00

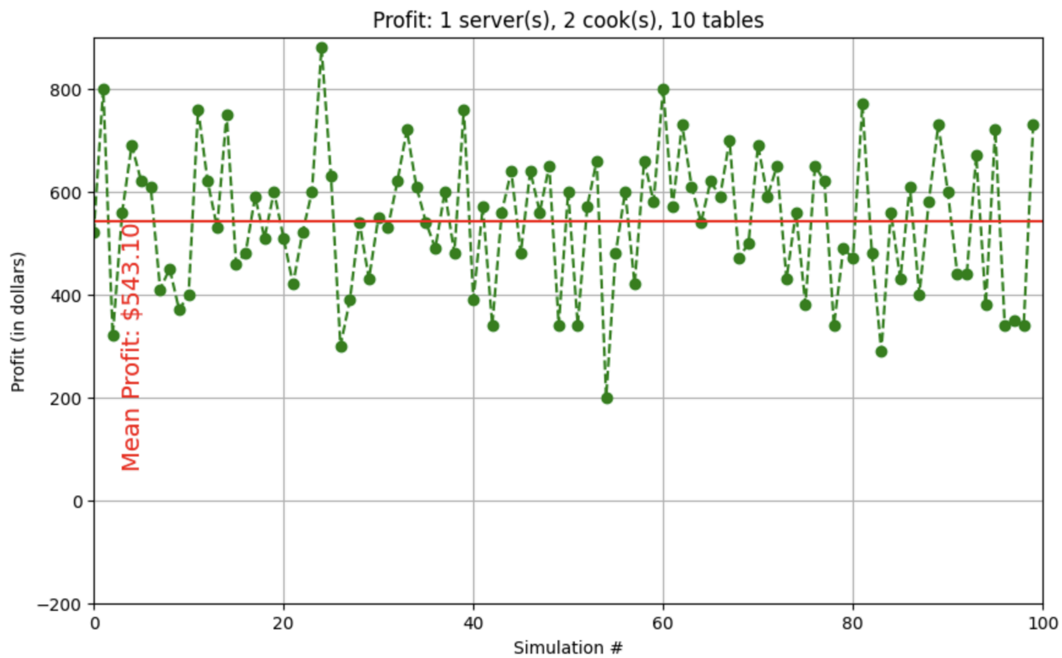

Figure 01. Scatterplot for the profitability of each simulation for 1 server, 2 cooks, and 10 tables.

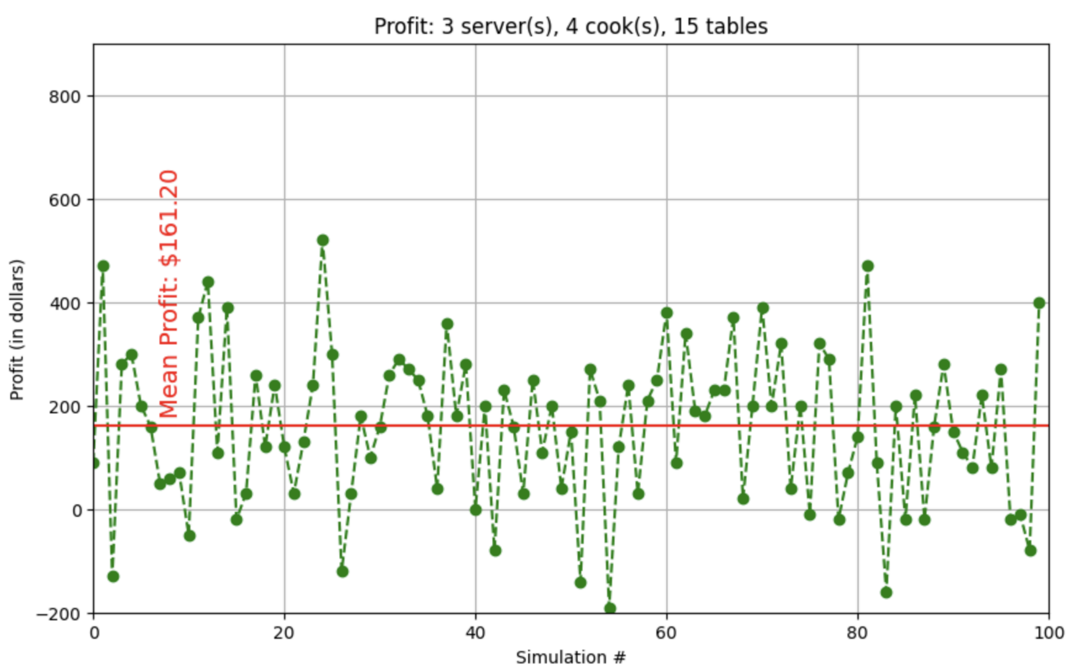

Figure 02. Scatterplot for the profitability of each simulation for 3 servers, 4 cooks, and 15 tables.

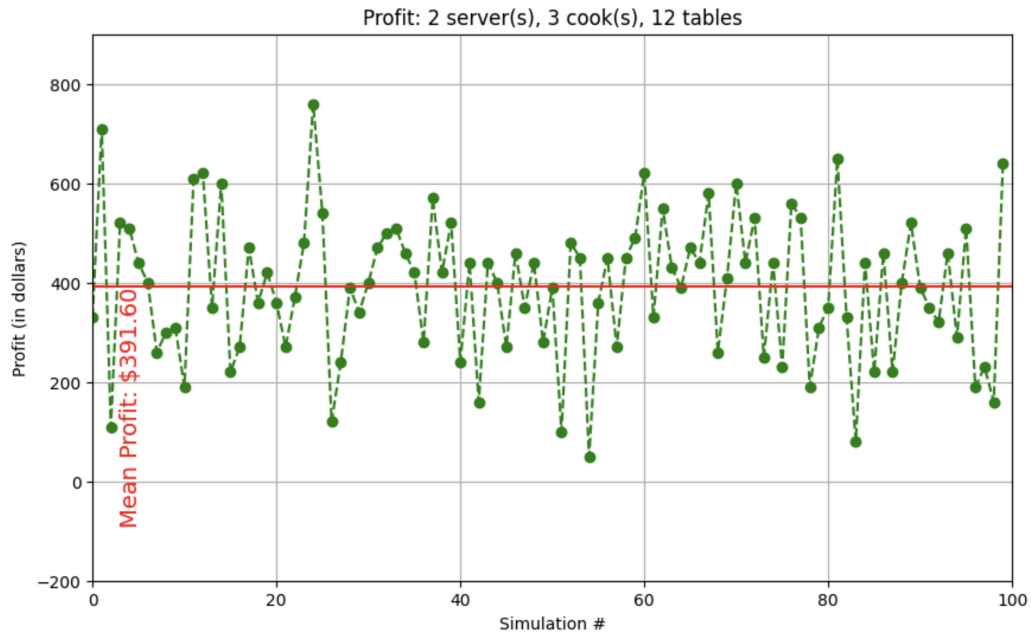

Figure 03. Scatterplot for the profitability of each simulation for 2 servers, 3 cooks, and 12 tables.

Conclusion

In this paper, we have created a SimPy discrete restaurant model that abstracts the critical processes involved in restaurant operations. The restaurant's profitability is directly linked to customer arrival rate, the number of employees (servers and cooks), and available resources for dining customers. Keeping other parameters constant, we varied staffing levels and the number of tables to determine the optimal profit. Based on this simulation using the parameters in Appendix 01, the optimal scenario is represented by XX servers, YY cooks, and ZZ tables.

Utilizing process simulations to make data-driven decisions on staffing levels can be powerful. As demonstrated with the discrete simulation restaurant, optimizing staffing levels can enhance efficiency, reduce costs, improve customer experiences, and unlock potential for increased profitability and growth. Factual data can determine the impact on profit and decision-making by measuring objective metrics, such as customers served per minute, providing an unbiased view of the situation. Businesses can use such metrics to make more informed decisions and improve operations. The idea is simple: increase service rate and decrease the need for limiting resources such as employees (servers and cooks) or facilities (tables). For example, ideas to increase the serving time include self-service kiosks and mobile ordering, which can improve efficiency and customer satisfaction and reduce customer waiting times during peak hours.

The developed model is not limited to a specific industry. It is a versatile solution that can be applied to various sectors, including Retail, Customer Service, and Hospitality, catering to any business whose operations can be modeled by a queueing model. Retail stores, call centers, and customer service departments often experience fluctuating demand based on seasons, holidays, and promotions. Simulations help determine the optimal staff needed during peak hours or busy periods. Businesses can allocate resources efficiently by modeling different scenarios while maintaining service levels. Hotels experience fluctuations in occupancy rates based on events, holidays, and tourist seasons. Monte Carlo simulations, a statistical technique used to model the probability of different outcomes, can optimize front desk staffing, housekeeping schedules, and restaurant staffing, balancing guest satisfaction with labor costs.

Furthermore, the Monte Carlo technique can be employed for sensitivity analysis and what-if scenarios, such as designing experiments to change the number of employees and resources. The sensitivity analysis can be expanded in future exercises to determine the results of serving patrons faster and determining demand over different scenarios, like days of the week, seasonality, and holidays. In all these cases, using actual data, such as direct information (collecting the service rate performance in the restaurant) or indirect metrics (determining service rate using payment invoice data), is crucial. This data should complement and feed the simulation to improve its accuracy and reliability, instilling confidence in its results.

Bibliography

Debjit, Roy, Eirini Spiliotopoulou, and Jelle de Vries. “How Data-Driven Decisions Help Restaurants Stay Competitive.” Harvard Business Review. October 03, 2023. https://hbr.org/2023/10/how-data-driven-decisions-help-restaurants-stay-competitive

Zhané, Jaya. “SimPy: Simulating Real-World Processes With Python.” Real Python. Accessed on June 01, 2023. https://realpython.com/simpy-simulating-with-python/

Zinoviev, Dmitry. PragPub February 2018. “Discrete Event Simulation.” The Pragmatic Bookshelf. Issue 104. February 2018. https://www.researchgate.net/publication/322949363_Discrete_Event_Simulation_It's_Easy_with_SimPy

Appendix 01. Basic simulation parameters

Parameter	Description	Value	Unit
NUM_SERVERS	Number of Servers	1	people
NUM_COOKS	Number of Cooks	2	people
NUM_TABLES	Number of Tables	10	tables
NUM_TAKEOUT_QUEUE	Max length of the take-out queue	5	consumers
NUM_DINEIN_QUEUE	Max length of the dine-in queue	5	consumers
DINE_IN_PROB	Dine-in Probability	50%	
TAKEOUT_PROB	Take-out Probability	50%	
SIMULATION_TIME	Simulation Time	480.00	min
MIN_PATIENCE	Number of Patience	1.00	min
MAX_PATIENCE	Number of Patience	2.00	min
MIN_SEATING_TIME	Max length of the seating time	1.00	min
MAX_SEATING_TIME	Max length of the seating time	5.00	min
DINEIN_SERVICE_TIME	Max length of the service time	5.00	min
TAKEOUT_SERVICE_TIME	Time to finish take-out service	3.00	min
TAKEOUT_SERVICE_RATE	Take-out service rate	0.33	orders/min

Parameter	Description	Value	Unit
PAYMENT_TIME	Time to finish the payment	1.0	min
ARRIVAL_RATE	Arrival Rate	0.20	consumer/min
COOK_TIME	Time to finish cooking	10.00	min
KITCHEN_COOK_RATE	Cook rate in terms of orders per minute	0.10	orders/min
SERVER_WAGE	Wage Rate for Servers	15.00	$/h
COOK_WAGE	Wage Rate for Cooks	15.00	$/h
REVENUE_PER_DINEIN	Revenue per Dine-In	30.00	$/order
REVENUE_PER_TAKEOUT	Revenue per Take-out	20.00	$/order
RANDOM_SEED	Number of Seed	42	
NUM_SIMULATIONS	Number of Simulations	100	