Cashier or Self-checkout? A cellular automata model to analyze grocery store checkout efficiency

Alaina Birney, Andrew Borah, Maya Griffith, Anika Hamby, Marit Scott

12/14/2023

Our Notebook:

MOCSFinalSelfCheckoutModel.ipynb

Abstract:

This paper presents a cellular automaton (CA) model designed to simulate the efficiency of using self-checkout versus traditional cashier methods in a grocery store. The model represents individuals with shopping carts or baskets full of items as cells, considering factors such as the number of items and item types in their cart. We chose to use cellular automata to model self-checkout and traditional cashier methods at a grocery store because we are interested in the queuing behavior of customers as well as their progression through the checkout process. We felt that CA were a great way to study this behavior because they provide a simple yet powerful framework for modeling complex systems with local interactions. CA can simulate the dynamics of customer movement and interaction in a discrete space, allowing us to analyze patterns and efficiency in the checkout process while keeping things relatively simple computationally.

Simulations are run with different distributions of product types and varying initial line lengths. Two experiments are run with this model: in experiment 1, the ratio of self-checkout to standard checkout machines is varied. In experiment 2, the most prevalent item in customers' carts is varied. Output metrics include the average number of customers served at cashiers and self-checkouts and the average number of items processed. The motivation for this model stems from the practical question of which checkout method is more efficient and its relevance to business decision-making, aiming to minimize customer wait times and enhance customer satisfaction, therefore our research question is which checkout method should a customer use? A synchronous discrete time and space CA model was used to run our simulations. This consisted of a 30 x 30 grid with traditional cashiers and self-checkout stations at predetermined locations. Based on the preliminary results of our model, the traditional cashiers had a higher average customer volume output as well as a higher average number of items processed. More refinement of the model will allow for more variation in simulations and improved outcome metrics, therefore more research is warranted.

Introduction:

In the contemporary retail landscape, optimizing checkout processes is crucial for ensuring customer satisfaction and business success. The choice between traditional cashier and self-checkout methods plays a pivotal role in shaping the customer experience. This paper explores a cellular automaton model developed to simulate and analyze the efficiency of these checkout methods in a grocery store setting. We chose to model this particular problem because cellular automata provide an efficient framework for simulating what can happen to an individual cell over a variety of conditions. This allowed us to easily implement several rule sets to simulate different real-life scenarios such as having a variety of items per customer or having a varied amount of people in the store or in line at the preferred checkout system.

A literature search for similar keywords and topics revealed some existing information available. *The Effectiveness of Self-Checkout, An analysis of market resistance, growth inhibitors, and suggested improvements* by Justin M. Edwards and Ryan S. Kenner shows that the time required to use self-checkout may exceed the time required to use a cashier-operated checkout. This finding was acquired through analysis of customer and employee survey responses, responses to questionnaires, observational data, and experiential studies of cashier-operated and self-checkout stations. Data sources included three grocery stores that were local to the researchers as well as five retail stores.

In *Prediction of Purchase Behavior of Customers in a Store by Cellular Automata,* Ryusuke Taniguchi et al. used cellular automata to model a grocery store. However, the goal of this study was to understand purchase behavior of customers as it relates to store layout in order to determine the most profitable layout for stores. Although the research question for this study differs from ours, the reasoning for using cellular automata to model the store is similar; both the model in this study and our model utilized cellular automata. Cellular automata are a good choice for studies such as this referenced study as well as our study because this type of model allows for assumptions to be made that make it possible to create a model of a complex system like a grocery store or checkout process while still producing valuable and variable results. This allows for a large application of situations in which the model can provide insight.

Society 5.0; a simulation study of self-checkout operations in a grocery store, Mykoniatis et al. Utilized AnyLogic simulation software to run a discrete event, agent-based models of a grocery stores in which customers arrive at checkout and select either self-checkout or traditional cashier. Unfortunately, it did not specify the rule set for line selection. Afterwards the customers unload, scan, bag, and pay. While this model has a lot of similarities to our work, it differs in its output. This study aimed to analyze throughput, which they defined to be the average time it took to get through the checkout process. This included both self-checkout and cashier and did not actually compare the difference between self-checkout and cashiers. This model would be a nice exploration for companies deciding to incorporate self-checkout machines or not, but not a comparison of the two methods.

As one last example, *A Markovian Queueing Model for the Analysis of User Waiting Times in Supermarket Checkouts, Reinaldo Morabito et al.* Set out to analyze the increasing problem of long waiting times at Brazilian supermarkets. It focused specifically on trying to determine the appropriate number of cashiers, by utilizing a simple Markovian model. This study did not only fail to compare self-checkout to cashiers, but also did not even incorporate self-checkout stations in their analysis.

None of the existing papers that we could discover through a literature review utilize cellular automata as a modeling tool so that a comparative analysis of the efficiency of self-checkout versus cashier operated checkout could be performed. This makes our work novel and an interesting thought experiment that could work as a motivator to more complex and accurate models.

<u>Methods:</u>

Implementation

Our model was a discrete time and space model with synchronous updates. As discussed previously, we chose to model our scenario this way because we are interested in the queuing behavior of customers and their progression through checkout stations. Our model consists of a 30 x 30 grid representing the grocery store checkout area to allow for sufficient space for the accumulation of queues, placement of checkout stations, and customer movement around these stations.

At each timestep, represented in our model as 5 seconds, customers in line will move forward into either a self-checkout or traditional checkout station. These customers move into lines at the same rate. This is not realistic to a normal grocery store, as there are people of many ages within a supermarket. For example, younger people may walk faster than the elderly. To make the model more realistic, we could make the customers flow at random rates. We decided to make everyone 'walk' at the same rate because we wanted to concentrate on the actual effectiveness of self-checkout machines, instead of the walks of customers. We also decided this because elderly people may be more likely to go to a regular cashier instead of the self-checkout machines, and we solely wanted to study the flow rate of people through each type of machine. If everyone was the same 'person', and only the item and item types in their carts were randomized, we could better concentrate on the effectiveness of self-checkout machines instead of the chances that they may get a slow walker in their line.

The number of items that a customer is purchasing is randomly distributed between one and fifty items. In reality, the number of items that customers purchase might follow a bell curve, skewed distribution, or even a random distribution like what was implemented. We chose to make the number of items random because a random distribution avoids introducing biases associated with assuming a specific pattern for the number of items purchased. Different grocery stores or regions may exhibit varying purchasing behaviors, and a random distribution provides a more neutral and unbiased representation of customer behavior. A random distribution also simplifies the model and analysis. Using a bell curve or skewed distribution would introduce additional complexity in terms of parameters and statistical considerations, potentially making it harder to interpret the results. It also allows for flexibility in the simulation, accommodating various shopping behaviors without the need to tailor the model to a specific demographic or locale. This makes the model more adaptable and applicable to different scenarios and settings.

Additionally, customers using self-checkout stations have a rate of procession that is dependent on the items in their cart; if their cart contains produce, they will have a slower rate of procession, and if it contains alcohol, the rate will be even slower. Introducing three types of items reflects the diversity of products typically found in a grocery store. The differentiation between standard, produce, and alcohol items allows for a more realistic simulation, considering the varying handling procedures, checkout times, and potential age verification requirements associated with each item type. We assumed produce and alcohol would take longer at self-checkout stations as they require manual input of produce codes, ID checks, and age approval from a clerk. This is realistic to real life; often during self-checkout, employees may have to ask for ID when buying alcohol, adding to the time it takes to check out your items. In addition to this, finding PLU codes will be much faster for a regular cashier, because they often memorize the codes. Someone who is not trained as a cashier and is using self-checkout will have to look at the product to find the PLU code, adding to the time that it takes to scan all of the items through self-checkout.

The customer's skill at self-checkout is uniformly distributed across all customers, and the only factor impacting the speed of checkout at these stations is the type of items within each customer's cart. In reality, customers exhibit a wide range of proficiency levels when using self-checkout machines. Factors influencing this skill set include age, technological familiarity, and prior experience with self-checkout systems. Additionally, individual comfort levels with technology and the ability to quickly and accurately scan items vary among customers. Therefore, assuming a uniform distribution of skills may not accurately reflect the diversity of customer capabilities. The decision to assume a uniform distribution simplifies the model, making it computationally more manageable. By isolating the impact of the type of items within a customer's cart as the primary factor influencing checkout speed, our model aims to emphasize the specific comparison between self-checkout and regular checkout processes based on item characteristics. This focus on items allows for clearer insights into the efficiency of each checkout method. It allows for a more controlled examination of the impact of item types on checkout speed without the added variability introduced by diverse skill levels. By assuming a uniform distribution of skills, our model aims to provide generalizable insights that can be applied to a broader range of scenarios and locations.

The rate of procession is constant for standard, cashier-operated stations, meaning it is the same regardless of the item types within the customers' carts. Assuming uniform skill levels among regular cashiers simplifies the model by removing the complexity associated with differentiating cashier skills. This is not comparable to real life, where some cashiers may have different skill levels than others. This decision enables a more straightforward comparison between self-checkout and regular checkout processes, with a focus on the impact of item types rather than cashier proficiency. We also assume that all cashier stations are staffed at all time points. With the rise of self-checkout machines, regular cashier stations are usually not all staffed. We made this assumption because it makes it easier to compare the different simulation results. If a randomized number of regular cashier stations were open, the results would then depend on how many of these machines were open, instead of solely the effectiveness of each type of machine.

We implemented our own neighborhood format where each customer in the back of the traditional checkout line looks to their right to move forward in line and once they are at the checkout, they look up to the belt and register. This can be visualized below in Figure 1.

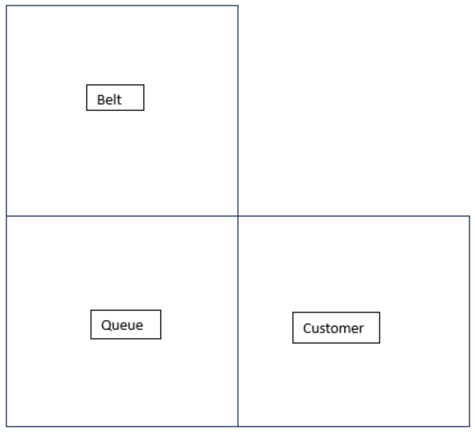


Figure 1. Visual representation of the neighborhood used at traditional standard checkout station

For the self-checkout stations, a singular line was populated to represent how people generally behave in grocery stores, by waiting in a single line until a station opens up.

A hard copy layout of our initial idea involving a random walk element in which shoppers would assess the states of the lines and gravitate towards their selected checkout station can be visualized below in Figure 2.

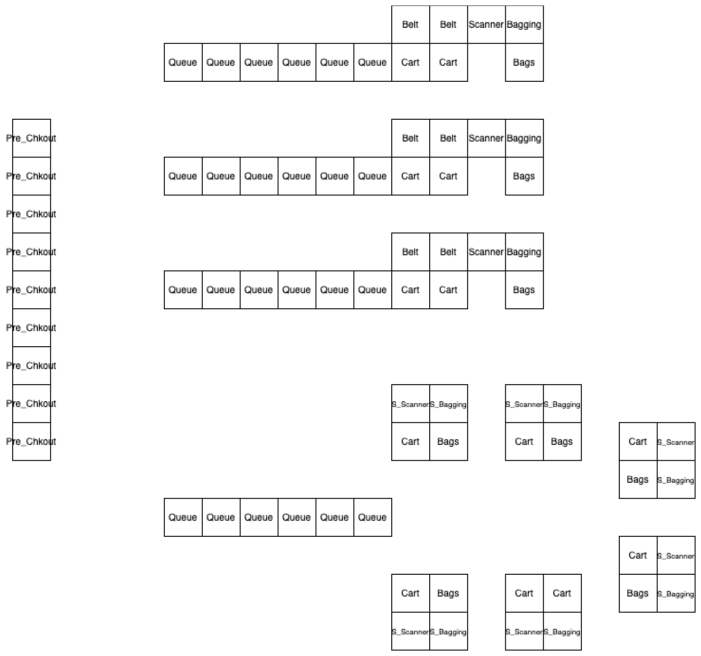


Figure 2. Initial layout of grid with random walk procession on the left side of the grid

Each customer is represented by a list of their items, in which standard items are stored as 0, produce items as 1, and alcohol items as 2. The final layout is visualized below in Figure 3.

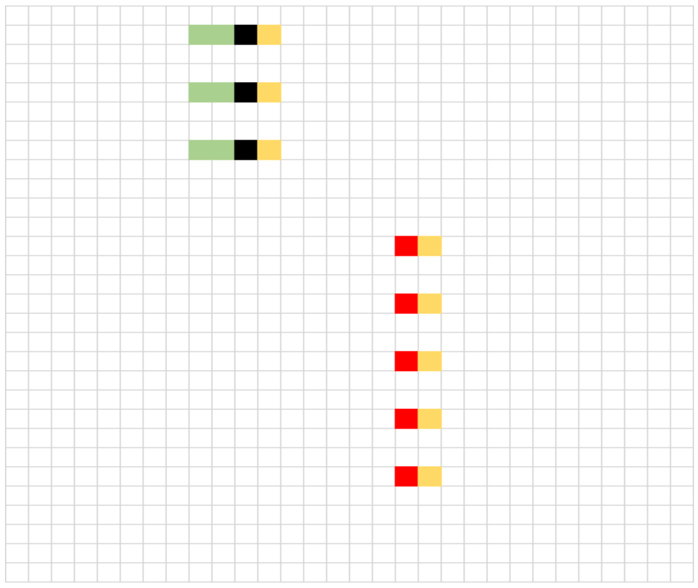


Figure 3. Green cells represent the belts at the traditional checkout stations. Black cells represent cashiers. Red cells represent self-checkout stations. All yellow cells represent bagging stations at either self-checkout or traditional cashier stations.

This layout retained the 30 x 30 grid but removed the random walk element of the model. A random number of undecided customers (between 0 and 3) are injected at each timestep. The choice to inject 0-3 undecided customers at each timestep acknowledges the unpredictability of customer arrivals and reflects real-world scenarios where the number of customers may vary throughout the day. Injecting a random number of undecided customers adds a dynamic and uncertain element to the model. It closely follows real-life conditions, where customers are constantly wanting to check out.

Customers choose lines by considering whether standard and self-checkout lines are busy (to be busy is to contain five people), the number of customers in each line, and the total number of items in each line. If multiple lines are not busy, then a line is chosen randomly. If only one line is not busy, the customer will go to that line. If all lines are busy, a customer will go to the line that contains the lowest total number of items. If all lines are full (to be full is to contain seven people), then the customer will wait until there is space in a line.

Considering factors such as line length, the state of the cashiers or self-checkout, and the total number of items in each line for customer decision-making replicates the decision process individuals undergo when choosing a checkout lane. Although, we do not take into account that some people in real life may prefer to go to a certain type of checkout, and not consider the other factors that were implemented in our method to choose a line. By including these criteria, the model accounts for the complexity of customer choices and the impact on overall system efficiency. The random selection when both lines are short introduces an element of unpredictability, mimicking real-world scenarios where customers might choose lanes impulsively. The choice to have customers wait to join a line when all lines are full was made for various reasons. First, this choice aids visualization; when lines become too long, it is difficult to see what is happening in the animation. Additionally, space is also limited in real life and this length could represent the length that a line can become before invading other spaces. However, the length that a line can become before invading other spaces is not often constant in reality because people can have different body sizes and levels of comfort regarding standing closely together. An arbitrary number was chosen for computational simplicity, but the model could be made more realistic by adding variability to the possible length of lines.

Limitations and Assumptions Specific to Experiments

Please see the section “Measurements” for additional details regarding the experiments.

- *Experiment 1: Three cashier stations and five self-checkout machines, five cashier stations and three self-checkout machines, or an equal number of each.*
    - The assumption that there would be three cashier stations and five self-checkout machines is somewhat accurate to the real world, where self-checkout machines often outnumber cashiers because they are increasingly cheaper operationally. We want to also test five cashier stations and three self-checkout machines to compare this configuration against the former scenario to analyze which scenario is more efficient. Lastly, we tested performance when there were an equal number of self-checkout and cashier stations to compare against existing scenarios.
        - The assumption reflects an exploration of various configurations of checkout lanes, a higher number of traditional cashier stations with a lower number of self-checkout machines and vice versa, as well as an equal number of each type

of machine. This allows for a comparative analysis of the impact of checkout lane distribution on overall system performance.

- *Experiment 2: Predominantly produce, alcohol, or standard items*
    - The assumption that all customers would be ten times more likely to have a certain product type within their carts is certainly not closely tied to reality. In reality, certain customers may be more likely to have a certain type of product in their cart, but this bias would not be uniformly applied to all customers. We made this assumption intentionally because we wanted to analyze the efficiency of standard and self-checkout relative to certain types of items while reducing computational complexity.
        - For a more realistic analysis, customers could continue to have randomly assigned item types in their carts and the time taken for customers to progress through self-checkout and standard checkout could be tracked relative to the distribution of items in customers' carts. This would allow for a comparative analysis of the efficiency of self-checkout and standard checkout relative to those item types to be performed while maintaining diversity in the distribution of types of products within customers' carts.

Measurements

The results were measured in several ways. We measured the average number of customers served at the standard, cashier operated stations and self-checkout stations, respectively as well as the average number of items processed at each of these groups of stations. These averages were found after 10 runs of the simulation. For general results, we display these average values of 10 simulations for various timesteps to show how the values change over time. For results specific to experiments, we display the final averages found after the simulation was run 10 times for 1,450 timesteps.

General Results:

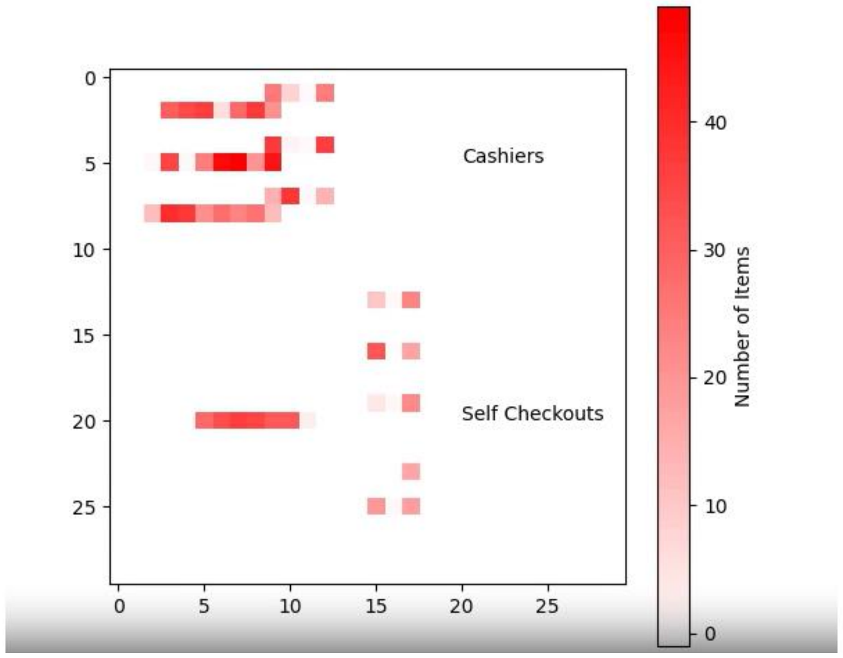


Figure 5. Representation of a timestep during a simulation.

Figure 5 above represents a frame in the animation of the actual simulation. As the color bar on the right of the figure depicts, the number of items at each cell is represented by varying shades of red with a darker shade representing more items (approaching the maximum number of items possible in a space, 50) while a lighter shade represents less items (approaching the minimum number of items possible in a space, 0). Additional specifics regarding the layout of stations can be found by referencing Figure 3. When the belt or self-checkout station has no more items, it will be filled with the next customer in the queue.

Figures 6 and 7 below show the results of running the checkout simulation over various lengths of time. The values seen in these figures represent the average values found at each timestep over ten runs of the simulation. It can be observed that standard, cashier operated checkout stations consistently serve more customers and process more items than self-checkout stations.

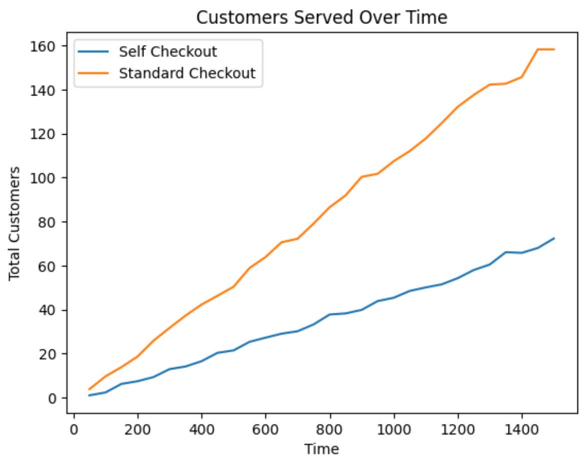


Figure 6: The average number of customers served over ten runs of 29 simulations of various durations, from 50 to 1450 timesteps (approximately five minutes to two hours).

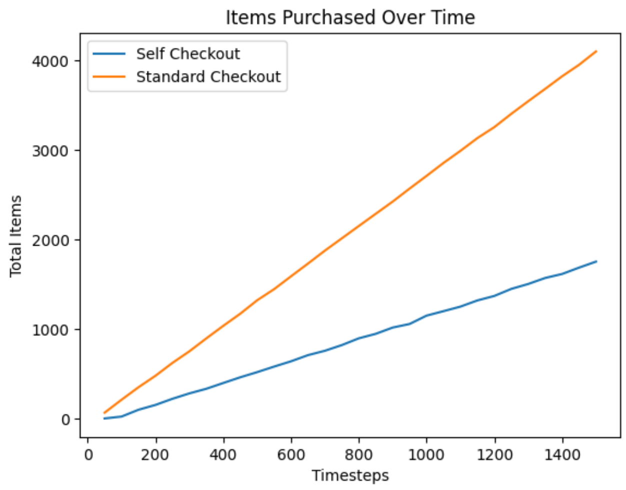
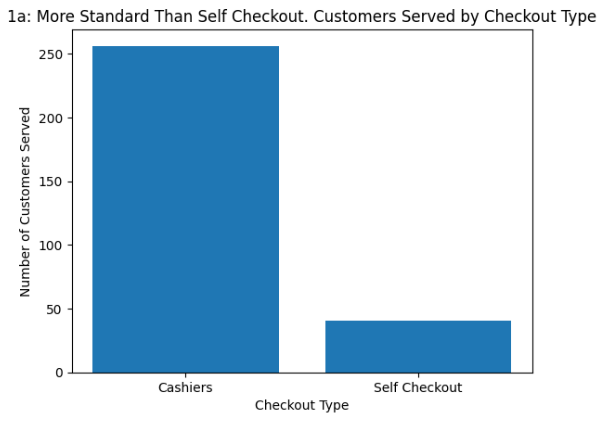
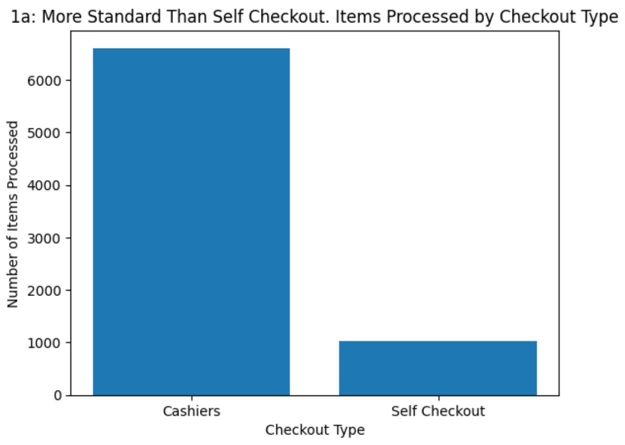


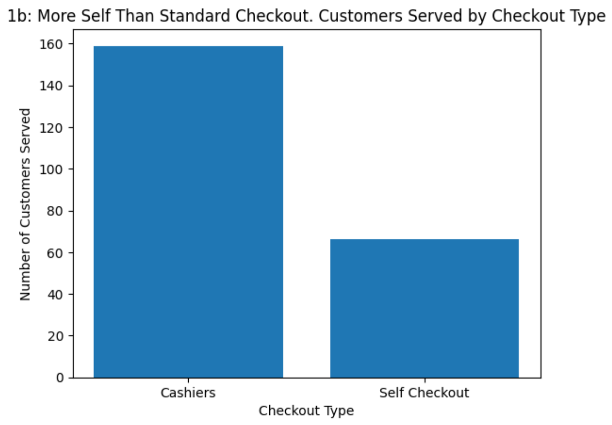
Figure 7. The average number of items processed over ten runs of 29 simulations of various durations, from 50 to 1450 timesteps (approximately five minutes to two hours)

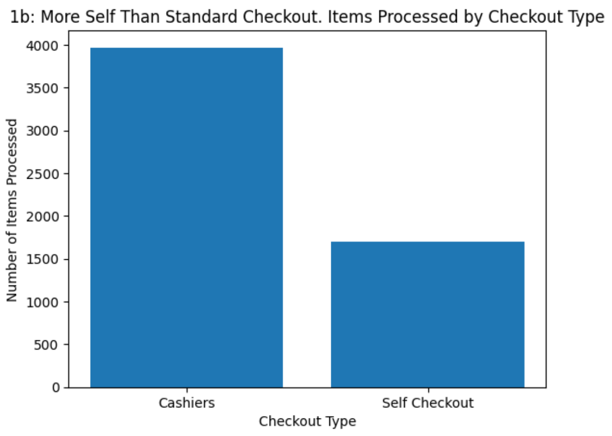
Experiment 1

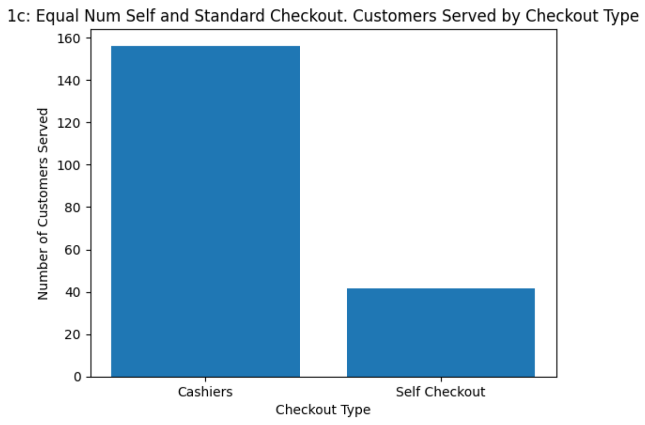
For experiment 1, we chose to vary the ratios of self-checkout to standard checkout machines. This experiment consists of three different scenarios: 1a, 1b, and 1c. In scenario 1a, there are 5 standard checkout machines and 3 self-checkout machines. In scenario 1b, there are 3 standard checkout machines and 5 self-checkout machines. In scenario 1c, there are 3 standard checkout machines and 3 self-checkout machines. Results for each scenario represent the average number of customers served and average number of items processed when the simulation was run 10 times for 1,450 timesteps.

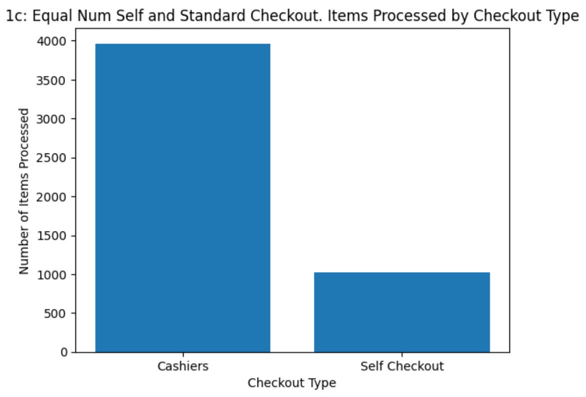








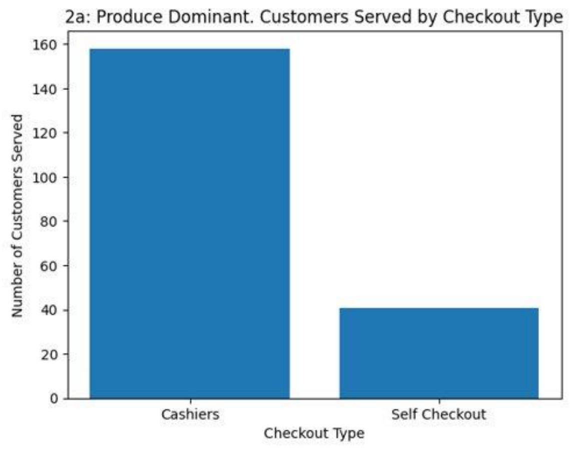


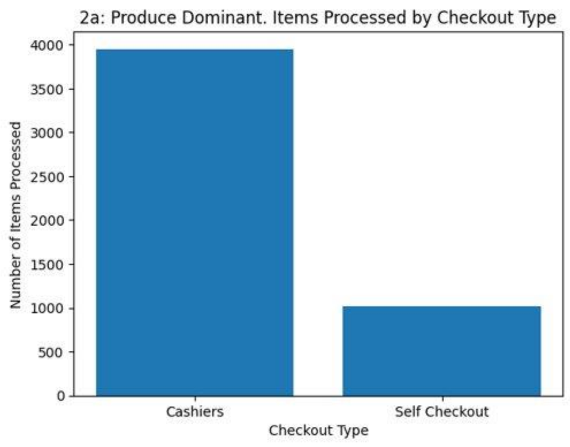


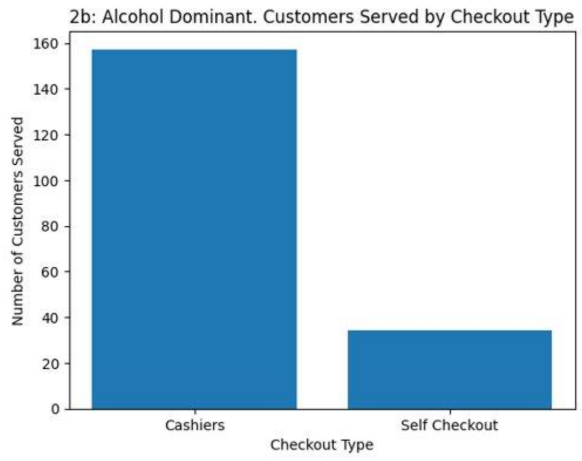
Across all scenarios, standard checkout stations consistently served more customers and processed more items than self-checkout stations. This could indicate that cashier-operated checkouts are generally more efficient at handling a higher volume of transactions. We saw the most significant difference in performance between self-checkout and standard checkout machines in scenario 1a where there were more standard checkout than self-checkout machines. We saw a decrease in this difference in scenario 1b where there were more self-checkout machines than standard checkout machines. This implies that increased availability of self-checkout stations may increase utilization but shows that they still underperform in comparison to standard, cashier operated checkout stations.

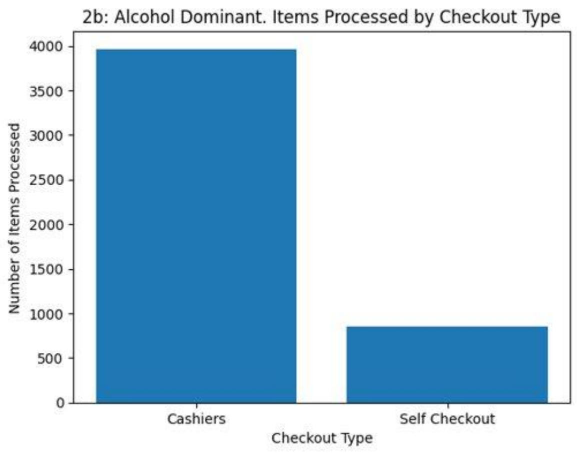
Experiment 2

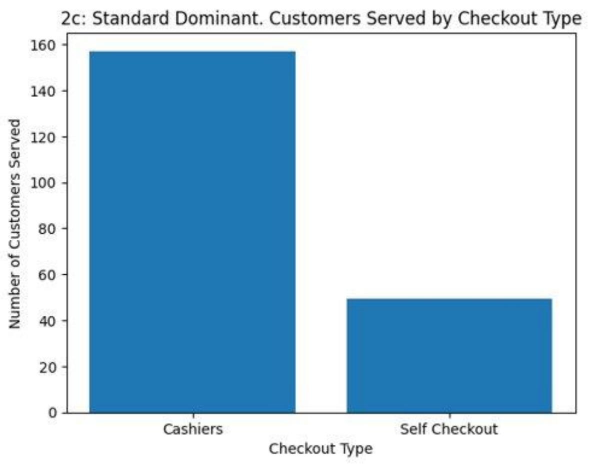
The second experiment, similarly to the first, consisted of three varying parameter situations. In the case of experiment 2 we chose to vary the odds of each item type to see if it had an overall effect. In the first scenario customers were ten times as likely to have produce in their carts or baskets compared to general items and alcohol. The second scenario in experiment 2 consisted of costumers who were ten times likelier to have alcohol, and in the last iteration customers were ten times likelier to have general items when compared to alcohol and produce. It is important to note that in the second experiment, there were 3 cashiers and 3 self-checkout machines for each scenario. The thought process behind this was to reduce any bias that may have been introduced from varying numbers of stations, which we observed in experiment 1. Results for each scenario represent the average number of customers served and average number of items processed when the simulation was run 10 times for 1,450 timesteps.

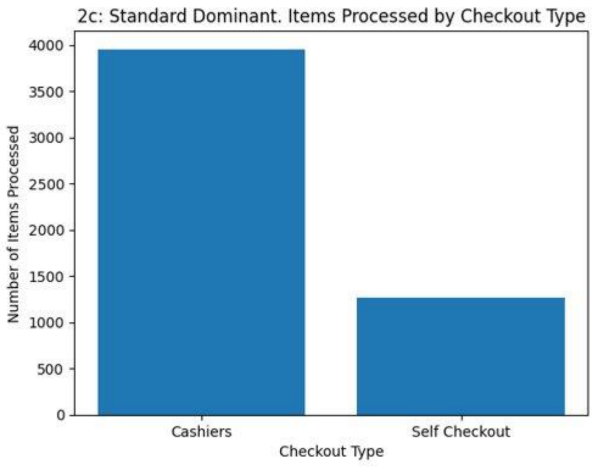












Looking at the data from experiment 2 we can pull out a couple of noticeable patterns. The most obvious is that the alcohol had the largest negative effect on both the number of customers served and the number of items processed. Produce was next, slowing down the checkout process for both the number of customers served and items processed. Lastly, an increase in general items had the smallest negative effect on checkout times. Another pattern to note is in all three derivations of experiment 2 the self-checkout was less efficient in both customers served and items processed.

Discussion & Conclusion

The preliminary data we saw in our work supports our hypothesis that a multitude of factors determine which checkout station is right for each customer. Given the inherent increase in time it takes to checkout with alcohol or produce, it is of no surprise that traditional checkout may be a better option. However, this could be impacted by the number of cashiers and self-checkout available, the other items intended for checkout, or other factors like the length of the line and the number of items in the carts of those in line. A limitation of our model were the assumptions we used to make the model computationally feasible in the time frame. This limited our ability to analyze scenarios such as staffing issues or limited numbers of customers in the store.

Our future work will include testing different scenarios of ratios of produce, alcohol, and standard items, and analyzing how fast either of the types of checkouts will complete checking out everyone in their line. One scenario would involve the customers choosing their checkout method based on the number and types of items they have. This may tell us whether there is a way for customers to strategize their approach to the checkout process based on what they are purchasing. We may also test the different numbers of self-checkout versus standard checkout and compare the output metrics over the varied parameters to see the effect of checkout types available. This would be a very interesting outcome to analyze as it would prove useful to anyone hoping to open up a store by providing direct feedback on the different ratios of checkout types.

Documentation:

Class Standard_Checkout:

active_cart: list representing the items currently being processed on the checkout cart.

active_belt: list representing the items currently on the active conveyor belt.

holding_belt: A list representing the items on the holding conveyor belt.

holding_cart: A list representing the items on the holding cart.

scanner: A list representing the items currently being scanned.

bagging: A list representing the items being bagged.

bags: A list representing the bags used for bagging.

queue: A list representing the queue of customers.

x_pos: The x-coordinate of where the standard checkout can interact with carts on the grid

y_pos: The y-coordinate of where the standard checkout can interact with carts on the grid

update_grid(grid): sets the grid configuration of the checkout area.

num_customers_served: Initialized to 0, representing the number of customers served so far.

customer_details: A list to store details about customers.

> *add_to_queue(self, customer):* Adds a customer to the queue for checkout.
>
> *no_cart(self):* Checks if there are no items either on the holding cart or holding belt, returning True if both are empty and False otherwise.
>
> *new_cart(self, cart):* Sets the items in the holding cart to the specified cart and updates customer details, including the ratios of regular, produce, and alcohol items in the cart.
>
> *print(self):* Prints the lengths of various queues and areas in the checkout system, including holding and active belts, scanner, bagging area, holding and active carts, and bags.
>
> *get_num_customers_served(self):* Returns the total number of customers served so far.
>
> *get_num_custs_in_line(self):* Returns the number of customers currently in the queue.
>
> *get_num_items_in_line(self):* Returns the total number of items in the queue.
>
> *update_grid(self, grid):* Updates the provided grid with the current state of the checkout system, including the lengths of holding and active carts, belts, scanner, bagging area, and bags.
>
> *timestep(self, grid):* Represents a time step in the checkout simulation, where items are moved between different areas based on the checkout process. It handles the movement of items from the holding to the active cart, from the holding to the active belt, from the active cart to the active belt, and from the active belt to the scanner and bagging areas. It also manages the completion of bagging and the transition of items from the bagging area back to the cart. The function updates the grid to reflect the current state after each time step.

Class Self_Checkout:

cart: A list representing the items on the holding cart.

scanner: A list representing the items currently being scanned.

bagging: A list representing the items being bagged.

bags: A list representing the bags used for bagging.

queue: A list representing the queue of customers.

x_pos: The x-coordinate of where the self-checkout can interact with carts on the grid

y_pos: The y-coordinate of where the self-checkout can interact with carts on the grid

has_paid: A boolean indicating whether the customer has completed the payment process.

produce_rate and *alcohol_rate:* Variables representing the number of extra timesteps needed to scan special items (produce and alcohol).

produce_check and *alcohol_check:* Variables used to enforce the slowdown rates for produce and alcohol scanning.

num_customers_served: Initialized to 0, representing the number of customers served so far.

customer_details: A list to store details about customers.

> *no_cart(self):* Checks if there are no items in the cart, scanner, bags, or bagging area, returning True if all are empty and False otherwise.
>
> *new_cart(self, cart):* Sets the items in the cart to the specified cart and updates customer details, including the ratios of regular, produce, and alcohol items in the cart.
>
> *print(self):* Prints the lengths of the cart, scanner, bagging area, and bags.
>
> update_grid(self, grid): Updates the provided grid with the current state of the self-checkout system, including the lengths of the cart, scanner, bagging area, and bags.
>
> get_num_customers(self): Returns the total number of customers served so far.
>
> timestep(self, grid): Represents a time step in the self-checkout simulation, where items are moved between different areas based on the checkout process. It handles the movement of items from the scanner to bagging, from the scanner to bags (considering special items), from the cart to the scanner, and the completion of the payment process. The function also manages the transition of items from bagging back to bags after payment. The *produce_check* and *alcohol_check* variables are used to enforce slowdown rates for scanning produce and alcohol items. The function updates the grid to reflect the current state after each time step.

General Functions:

fill_cart(customers): Generates a random list of items for a customer's shopping cart, where each item is assigned a type (standard, produce, or alcohol). The function appends the generated cart to the list of customers and returns the updated list.

get_self_checkout_num_items(self_checkout_line): Computes the total number of items in the self-checkout line by summing the lengths of the carts for each customer in the line.

choose_line(customers, standard_checkouts, self_checkouts, self_checkout_line): Decides which checkout line (either standard or self-checkout) a customer should join based on several criteria. It considers whether standard and self-checkout lines are busy, the number of customers in each line, and the total number of items in each line. The function randomly selects a line if both standard and self-checkout lines are relatively short. If only one type of line is not busy, it chooses that line. If all lines are busy, it selects the line with the fewest total items. The chosen line is then updated with the customer.

count_item_type(cart, item_type): Returns a count of the specified item_type within the cart (which is a list).

animate(frame): Manages the dynamic progression of a checkout simulation. It generates new customers with random items if the checkout queue is empty, adds customers to the queue if the total is below a threshold, advances the checkout process through a time step, and updates the visual representation of the simulation for animation purposes.

References:

1. Kenner, R., & Edwards, J. (2004). *The effectiveness of self-checkout, an analysis of market resistance, growth inhibitors, and suggested improvements.* Worcester: Worcester Polytechnic Institute.
2. R. Taniguchi, Y. Ohtaka and S. Morishita. (2015). *Prediction of Purchase Behavior of Customers in a Store by Cellular Automata*. IEEE.
3. Mykoniatis, Konstantinos & Shirzaei, Samira & Katsigiannis, Michail & Panagopoulos, Athanasios Aris & Deb, S. & Potter, T. & Angelopoulou, Anastasia. (2020). *Society 5.0: a simulation study of self checkout operations in a grocery store.*
4. Morabito, Reinaldo & de Lima, Flavio. (2004). *A Markovian Queueing Model for the Analysis of User Waiting Times in Supermarket Checkouts.* International Journal of Operations and Quantitative Management.