*Proceedings of the 2021 Winter Simulation Conference*
*S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.*

# CUSTOMER PATH GENERATION SIMULATION FOR SELECTION FROM PROPOSED GROCERY STORE LAYOUTS

Kimberly Holmgren

Georgia Institute of Technology
North Ave NW
Atlanta, GA 30332, USA

## ABSTRACT

Before a grocery store opens, key operational decisions must be made with no historical data. One important decision is how to optimally lay out the store to maximize consumer spending. This work reviews existing literature on simulation to optimize grocery store layout, uses computer vision techniques to transform a store diagram into a digital representation, and applies simulation methods to approximate which of several layouts proposed by a store designer would result in the highest amount of impulse purchasing. Output analysis methods are used to compare these results to determine whether one design outperforms the others.

## 1 INTRODUCTION

Conventional grocery stores typically have very low profit margins, averaging about 2.2% in the U.S. (Campbell 2020). In order to maintain profitability, several decisions can be made to decrease costs or increase sales. One method to increase sales is to choose a store layout to increase customer impulse purchases. Impulse purchases are unplanned purchases, and are viewed as an opportunity to increase sales over what a customer was planning to spend on commodities through factors the store can control. Impulse purchases are said to account for 30-50% of sales, and 3 out of 4 shoppers report making some purchasing decisions in-store (Kollat and Willett 1967; University of Southern California 2020). ”Must-have” or commodity purchases are items like eggs or toilet paper that a customer plans to purchase regardless of factors the store controls. Impulse purchases might include a pack of gum at the checkout aisle or a refrigerated beverage on a hot day. Whether the customer buys these items is heavily influenced by if they see them. Hoch and Loewenstein (1991) suggest that physical proximity is the strongest inducer of reference-point shifts, which are sudden desires that result in unplanned behavior. Therefore, a longer path through the store is desirable as it provides more opportunities for stimulating additional impulse purchases, which drives revenue.

When a new food co-op or grocery store plans to open there is an opportunity to incorporate data-driven decision making early to increase impulse purchasing and revenue. Importantly, at this phase decisions can still be made and implemented with minimal cost as there has not yet been significant investment into creating the physical layout. However, there is minimal data on customer behavior, product selection, or product placement, which is required to apply existing methods. Typically, one or several floorplans are created by a professional with knowledge of the planned inventory, available space, and target market. At this stage, the floorplan is defined as the block layout and the assigned department label. An example can be seen in Figure 1. This method accepts proposed block layouts and uses simulation techniques to recommend which floorplan is likely to be most profitable by using average customer path length as a proxy for impulse shopping. It contributes to the existing literature by providing an earlier stage solution for when limited data is available.

**978-1-6654-3311-2/21/$31.00 ©2021 IEEE**

## 2 LITERATURE REVIEW

There are two categories of related research on simulation to improve the profitability of a grocery store layout. Many papers aim to optimize grocery-store layout to stimulate impulse shopping, but require a significant amount of information to be available, such as product placement. Bhadury et al. (2016) apply the p-dispersion model to spread common "must-have" items across the store, thereby maximizing the customer's exposure to impulse items. Significant improvement was achieved using this method, but the scope and problem definition differ from that of selection from a fixed set of store layouts at the department-level before product placement decisions are made. Ozgormus and Smith (2020) propose a method for increasing revenue while considering adjacency preferences and validate it on two locations of the Turkish grocery chain Migros. Again, large improvements were achieved (3-4% increase in revenue), but the problem definition varies from the one described here; in that problem statement the layout is fixed and only the contents of those fixed departments vary. Finally, Dorismond (2019) outlines a simulation-based tool to guide periodic changes in a supermarket layout and an approach to dynamically position promotional products in a retail store. There are several interesting findings, but again the block layout is assumed to be fixed and the customer path and shelf allocation are explored. None of these solutions apply to the problem of comparing several proposed layouts because they iterate on a fixed layout or require data that's unavailable during the building stage. However, several of the strategies used in those simulations are considered.

There are also many approaches to simulate the path a customer will take within a grocery store. Larson, Bradlow, and Fader (2005) use a proprietary PathTracker®system which affixes RFID tags to grocery carts to study common paths. The paths identified are specific to the store layout where the data was collected, and cannot be applied to other layouts arbitrarily or used to collect data for layouts which do not yet exist. Hui, Fader, and Bradlow (2009) note the similarity of grocery store shopping to the traveling salesman problem (TSP) and study the ways customer paths systematically differ from the optimal TSP path. They found that customers deviate from the optimal path on average by 28% (ranging from 5 to 95%). The first type of deviation, order deviation, is when customers do not visit the next-closest item to where they currently are. This averaged 3% of the deviation. The second type of deviation, travel deviation, is when customers visit the next-closest item, but take a path other than the shortest path to get there. This averaged 69% of the variation. Order deviation is strongly correlated to basket size, while travel deviation is uncorrelated to basket size. For this reason, in this simulation for the purpose of estimating additional purchases only order deviation will be considered and customers will take the optimal path between items. Several papers use a one-step-ahead approach in which a customer selects the next item from their list with a probability inversely proportional to its distance from their current location (Bhadury et al. 2016; Dorismond 2019). That approach will be adapted for this simulation, but is not a complete solution to the problem of estimating relative profitability of several proposed layouts.

## 3 METHOD

The following is the proposed method to select from several proposed store diagrams from a planner by simulating customer walking path length as a proxy for impulse purchasing, which correlates with profitability. This method is for early-stage planning purposes, when limited data is available but changes to the block layout are still possible and cost-effective.

1. For each proposed store diagram in the format shown in Figure 1, parse the diagram using computer vision techniques to map pixels to department names.
2. Obtain the Instacart grocery dataset and create a map from the Instacart department labels to the proposed store department labels
3. For each customer $c_i$ in $c_1$, ..., $c_N$:
   (a) Create a shopping list $l_i$ for $c_i$ by sampling randomly from the Instacart dataset.
   (b) For each store layout $S_j$ in $S_1$, ..., $S_M$:

   i. Map each item $item_k$ from $l_i$ to a physical location in $S_j$ by randomly selecting a pixel associated with the department the item belongs to, labeled ($x_k$, $y_k$).
   ii. Set the customer to the entrance point of the store. Use a heuristic approach detailed in Dorismond (2016) to determine the order of visitation of locations in the store. When there are no items left, append the checkout department as the last stop.
   iii. For each movement between ($x_k$, $y_k$) and ($x_{k+1}$, $y_{k+1}$) track both the Euclidean distance and the A* path distance the customer would travel.
   iv. Return the total Euclidean distance traveled by customer $c_i$ in store $S_j$ as $E_{j,i}$ and the total A* path distance traveled as $A_{j,i}$.

4. For each pair of stores ($S_a$, $S_b$) use paired confidence intervals to compare $E_{a,*}$ and $E_{b,*}$ as well as $A_{a,*}$ and $A_{b,*}$.

Each step in this method is discussed in further detail below.

### 3.1 Parse Store Diagram

The first task in simulating the customer flow and purchasing activity in a store is translating a document representing the store's layout into a representation that can be used by the simulation. A simplified example of a floor layout plan is shown in Figure 1. There is a single column which contains a legend mapping colors to the name of the product type. The location where items of that type are stored is entirely shaded in the same color. There may be some additional features of the diagram, such as dimensions, bathrooms, and more, which are ignored.

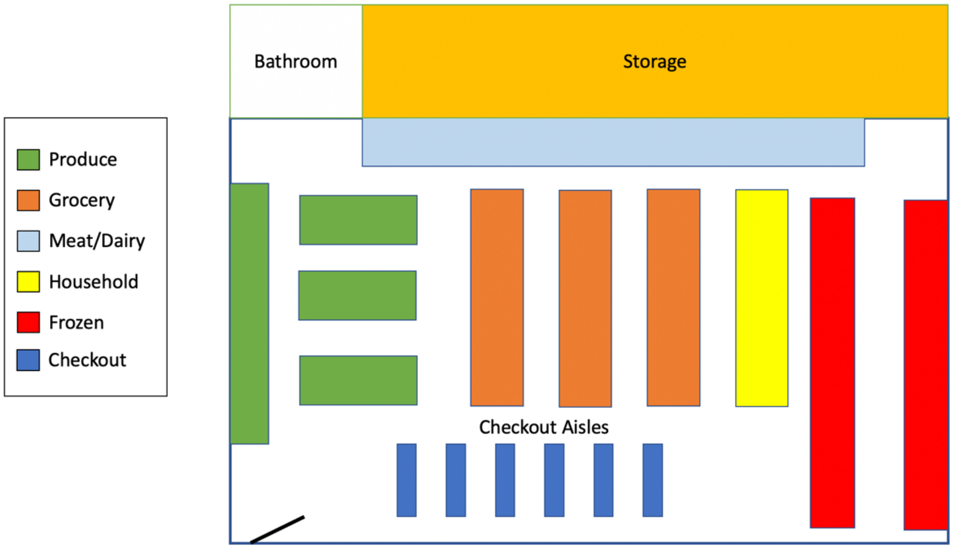

Figure 1: Example of a grocery store layout.

### 3.2 Parse Store Legend Colormap

To be able to map regions in the store to items a simulated customer is interested in, the legend should be parsed to map an RGB value to a department name. This is achieved as follows:

1. The legend is saved as a separate image, containing one column of color blocks matched with the aisle name per line
2. Use Tesseract OCR (Optical Character Recognition) to read all text from the image

3. Split on the newline delimiter and automatically clean up special characters. Results of OCR are typically not perfect, and manual cleanup may be required as well.
4. Identify unique colors in the legend image which appear in more than .05% of the pixels in the legend. Use KMeans clustering algorithm since most colors have variations in the pixel-level RGB values.
5. Sort the colors in the order they appear in the image, top to bottom, and assign them to the department names.

Other methods may achieve these tasks automatically and with lower error, but this step reduces the manual processing in mapping RGB values to department labels. This approach assumes black, white, and gray are not used as department colors and no colors are used in large amounts in the legend besides department colors. High and low ranges are specified as well to provide a flexible threshold for images where the borders have some blending. Example output of this step can be seen in Table 1.

Table 1: Sample output of legend parsing.

| Department Name | Low RGB | RGB Centroid | RGB High |
|---|---|---|---|
| produce | 110, 155, 69 | 125, 170, 84 | 140, 185, 99 |
| grocery | 208, 114, 52 | 223, 129, 67 | 238, 144, 82 |
| meat/dairy | 178, 199, 220 | 193, 214, 235 | 208, 229, 250 |
| ... | ... | ... | ... |

### 3.3 Preprocess Store Layout

The first step for the store layout processing is also to save it as a separate image, containing only the floorplan that a customer could walk through. The image is resized to to 500x500 pixels to standardize maximum path size and reduce the search space. Since items can only be placed on the perimeter of any aisles, it's necessary to find the coordinates of the perimeter for all departments. For each department the color mask created from the high and low values in the legend is applied to filter out pixels that are not part of the section. Then, a basic edge detection algorithm is applied to identify the perimeters. The edge pixel locations are stored as options for item placement for each department. A buffer of 10 pixels is set to indicate that products cannot be placed on the perimeter of the store. For example, the leftmost produce section will not have products on its leftmost perimeter, because that is against a wall and inaccessible to the customer.

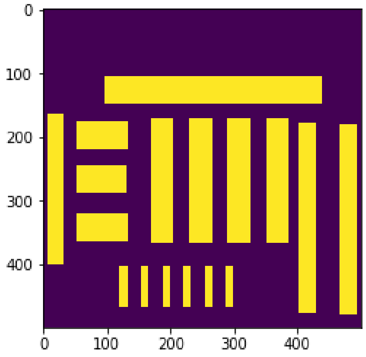

Figure 2: Walking path mask, purple denotes where customers can walk, yellow denotes obstacles.

Finally, a combined mask of all departments is created to denote where customers can and cannot walk for the path generation algorithm. This is shown in Figure 2.

## 3.4 Create Customer Shopping Lists

There are a few options to create customer shopping lists. One option is to create customer profiles based on clustering analysis, and create shopping lists for each profile (Bhadury et al. 2016; Larson et al. 2005). Another is to use association rule mining to identify rules for products customers typically buy together and create lists based on those rules (Ozgormus and Smith 2020). The simplest option, which is used here, is to directly use customer purchase lists. This limits the flexibility of the customers and the variability introduced into the simulation, which now mostly comes from path choice. This limitation is surpassed by using a large dataset. A more complex customer list creation mechanism could be a future step, especially as more information about expected customers becomes apparent since the profiles or rules can be adjusted to be more realistic. The dataset used here is the Instacart Online Grocery Shopping Dataset 2017 (Instacart 2017). During this work the website was temporarily unavailable, so the data was obtained through Kaggle (Psparks 2017). The dataset was chosen because of the large number of orders (around 3 million), the pre-existing categorization into departments, and, most importantly, because each order does not reflect impulse buys that were triggered by physical proximity. Some amount of impulse buying may occur due to recommender systems through the web app, and it is unclear whether that was a feature of the site prior to 2017. However, the Instacart orders should reflect a customer's "must-have" purchases more accurately than in-store data, which would also reflect impulse buys. To apply the Instacart dataset to a different categorization of departments, a mapping must be created between the departments in the dataset and those in the proposed layout legend. While there are several departments that are common between most grocery stores like produce and dairy, the categorizations may not exactly align. For the example grocery store layout in Figure 1, the categorization is shown below in Table 2.

Table 2: Mapping from example legend categories to Instacart Dataset categories.

| Example Departments | Instacart Departments |
|---|---|
| Produce | Produce |
| Grocery | Bakery, International, Beverages, Dry Goods Pasta, Bulk, Pantry, Breakfast, Canned Goods, Snacks |
| Meat/Dairy | Meat, Seafood, Diary Eggs, Deli |
| Household | Other, Pets, Personal Care, Household, Babies |
| Frozen | Frozen |
| None | Alcohol, Missing |

The categories are represented numerically in a table. Products in departments mapped to "None" are removed from orders. These department mappings are extended to product and order mappings. To generate a customer grocery list we can simply query an order from the Instacart dataset with the remapped columns to apply to the grocery store layout. The training set used from Kaggle contains 131,209 unique shopping lists. An example list is shown in Table 3.

The checkout section is always added to a customer shopping list as well, and it is always reserved to be the last item.

## 3.5 Translate Shopping List to Customer Path

Since item location within a given department is unknown, it is treated as a random location within the department. This means even if customer $c_1$ and customer $c_2$ have the same product on their shopping list they may find it in different locations within a proposed layout layout $S_1$. Similarly, if customer $c_1$ goes through $S_1$ twice, she may find the same item at different locations. In this way, there is variation introduced into the layout so no assumptions are made about where a product is placed. The coordinates for each product on a customer's shopping list are chosen randomly from the perimeter of the associated

Table 3: Example Instacart shopping list with remapped departments.

| Order ID | Product Name | Mapped Dpt |
|---|---|---|
| 1260810 | Organic Blueberries | produce |
| 1260810 | Chicken Maple Breakfast Sausage | frozen |
| 1260810 | Fridge Pack Cola | grocery |
| 1260810 | White Giant Paper Towel Rolls | household |
| 1260810 | Half Baked® Ice Cream | frozen |
| 1260810 | Cinnamon Raisin Swirl Pre-Sliced Bagels | grocery |
| 1260810 | 100% Whole Wheat Cinnamon with Raisins | grocery |
| 1260810 | Original Cream Cheese | meatdairy |
| 1260810 | Blueberry Beet and Brown Rice Cakes | household |
| 1260810 | Brownie Batter Core | frozen |
| 1260810 | Apple, Juicy Red, Family Pack | produce |
| 1260810 | Organic White Cheddar Macaroni & Cheese | grocery |

department in the store layout. Two examples of generating coordinates from the same grocery list are shown in Figure 3, where item placement coordinates are shown in purple.

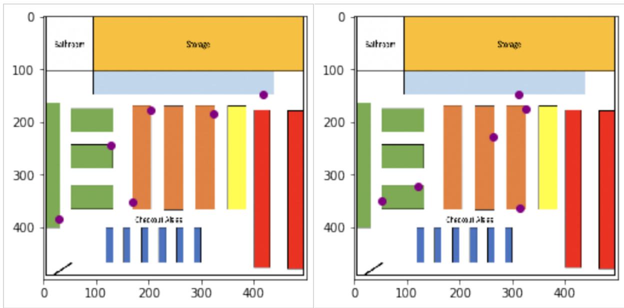

Figure 3: Two instances of the placement of products from the same underlying grocery list.

After the item coordinates are determined, they must be ordered by when the customer visits each. Starting coordinates must be specified for the path generator. In this case the customer starting point is the bottom left corner where the doorway is shown. From there, the approach proposed by Dorismond (2016) is used. A diagram is shown in Figure 4:

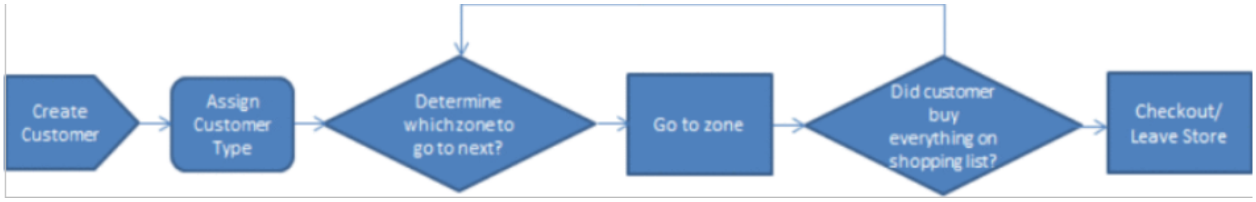
Create Customer
Assign Customer Type
Determine which zone to go to next?
Go to zone
Did customer buy everything on shopping list?
Checkout/ Leave Store

Figure 4: Diagram from Dorismond (2016) for one-step-ahead path generation.

At this point, the customer has been created and the customer's shopping list has been assigned. The decision for which zone to visit next is determined by calculating the Euclidean distance between the current location and all future points to visit. The Euclidean distance is chosen because it is quick computationally and gives an approximation for the customer's travel distance. The probability vector **Pr** is computed with Equation 1.

$$\begin{aligned}\mathbf{P} &= \left(\frac{1}{\mathbf{D}}\right)^n \\ \mathbf{Pr} &= \frac{\mathbf{P}}{|\mathbf{P}|}\end{aligned} \tag{1}$$

Here **D** is the vector of Euclidean distances between each unvisited point **C** and the current point $\mathbf{c} = [c_x, c_y]$, computed as $|\mathbf{C} - \mathbf{c}|$. The parameter $n$ is selected based on the expected distances for the image resolution to create desired behavior. For this experiment a value of 5 was chosen.

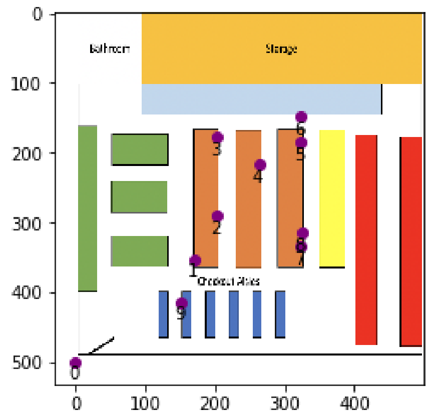

Figure 5: Customer visit locations denoted in purple, overlaid with assigned order.

Based on those probabilities, the next point is selected and the current point is removed from the list of points to visit. Once the shopping list is empty, a random checkout line is selected and the customer moves there to leave the store. The results of ordering a coordinate set is shown for an example set of points in Figure 5.

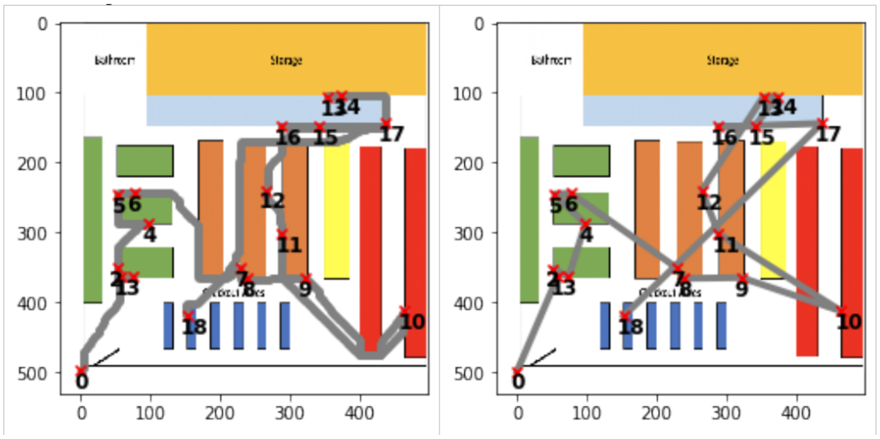

Figure 6: Customer path with A* shown on the left, customer path using Euclidean distance on the right.

The last step of path generation is to generate the path between these ordered points. One approach is to use the Euclidean distance, and ignore the physical obstruction of the aisles. This is used in Bhadury et al. (2016) and others. Another approach is to use a path generation algorithm such as the A* algorithm, a common successor to Dijkstra's algorithm which uses heuristics to achieve high performance in less time (Hart et al. 1968). However, it's still far less efficient than the simple computation of the Euclidean distance, and using the A* algorithm increases processing time. Figure 6 shows the same set of ordered coordinates to be visited, and the path the customer takes using A* as compared to the shortest Euclidean path.

### 3.6 Summary

With all of these components together, there is now a way to simulate customer path distance through a store layout. The legend is parsed from an image to identify the department names matching each color. A store map is parsed from an image to represent physical obstructions and the outlines of each department are stored for potential product placement. Customer shopping lists are generated using the Instacart Grocery Online Shopping Dataset of 2017, which is remapped to the departments existing in the store layout plan. Finally, a customer's path is generated by randomly assigning locations within a department perimeter to each product, ordering the visitation of those locations using a one-step-ahead method, and measuring the distance between points with either the Euclidean distance or the length of the path generated by the A* algorithm.

## 4 RESULTS

Three sample layouts were created to obtain results on the task of choosing between several proposed layouts. Figure 7 below shows these three layouts. In all of them the legend remains the same. To evaluate each layout, 1,000 customers are created, and each individual's path is tracked for the same grocery list through all of the layouts. The output is the path distance. Paired confidence intervals are used to compare the layouts and determine whether one is substantially better than the others in terms of mean customer path.

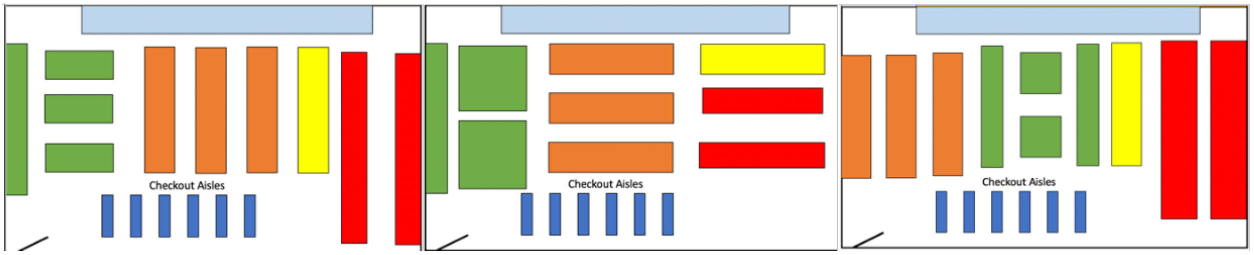

Figure 7: Three proposed layouts for a grocery store. The figure on the left, Layout 1, is the one used throughout the report. In Layout 2 in the center, many of the aisles are rotated to a horizontal orientation and the produce department shape sightly changes. On the right in Layout 3 the position of the grocery aisles switches with the produce section, and the produce section layout is altered.

The results are shown in Table 4 below. Using the Euclidean path distance, Layout 1 appears to generate the longest average customer path. Using the A* path distance, Layout 3 appears to generate the longest average customer path, with Layout 1 second.

Table 4: Average distance over 1,000 paired samples of customer walking paths.

| | **Euclidean Path Distance** | **A* Path Distance** |
|---|---|---|
| **Layout 1** | 1245.85 | 1859.86 |
| **Layout 2** | 1206.87 | 1664.74 |
| **Layout 3** | 1205.3 | 1881.03 |

Since these samples were taken in a way that introduces dependence (the same customer moves through all three layouts), paired confidence intervals are used to evaluate the difference between the means of each system. The values of each distance metric for each layout are approximately normal, which allows this approach. Results for the A* walking path are shown in Figure 8.

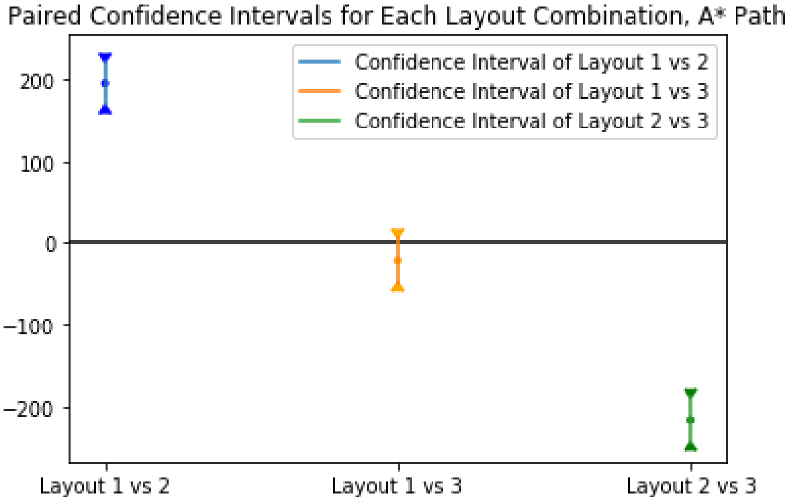

Figure 8: Results of comparing layouts pairwise by A* walking path.

There is a distinguishable difference between Layouts 1 and 2. Since the confidence interval with $\alpha$=.05 lies entirely above 0, it's been shown with 95% confidence that Layout 1 generates a longer average walking path than Layout 2. The confidence interval comparing Layout 1 to Layout 3 includes 0, so we cannot conclude that one layout is significantly better than the other, although in this sample Layout 3 generated a higher average walking path. Finally, the confidence interval comparing Layout 2 to Layout 3 is entirely below 0, so Layout 3 generates a longer walking path than Layout 2.

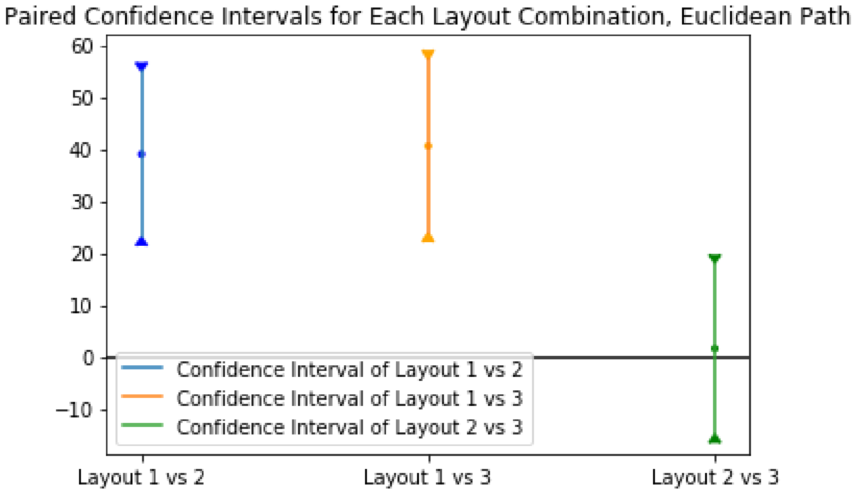

Figure 9: Results of comparing layouts pairwise by Euclidean walking path.

This analysis is repeated using the Euclidean path measure and results are shown in Figure 9. In this analysis, Layout 1 is shown to be better than Layouts 2 and 3 with 95% confidence, and Layout 2 and 3 are indistinguishable. This result differs from that found with the A* path, which is more representative of a customer's true walking path. This is an important finding because other methods evaluate their layout with the Euclidean distance, which is now shown to not always correspond to the walking path when accounting for obstacles.

Another way to compare many layouts is to use a selection method adopting the indifference-zone approach. This means comparing the best system, or subset of systems within the same performance zone such that we are indifferent between those systems. This method is not used here, but may be useful in comparing more than three layouts.

In addition, a brief timing analysis was completed. The time to compute one customer's path through each of the 3 layouts using both the Euclidean distance and the A* path distance averaged 37.8 seconds, with a standard deviation of 16.75 seconds. The paths of the 1,000 customers for each of the three layouts from this simulation took 10.6 hours to calculate. Improvements in processing speed could be achieved by using only the Euclidean distance. To create 1,000 customer paths and calculate distance using Euclidean distance only, it takes only 7 seconds, or on average .007 seconds per customer. The tradeoff is that Euclidean distance is a less accurate representation of the walking path. Another option is to constrain the graph search space by segmenting the image into overlaid nodes in an algorithmic way or in a crude way (like further downsampling the image). Finally, other path generation algorithms could be explored to increase efficiency.

Notably, it is very difficult to validate this solution in a real world setting because only one of several proposed layouts will be built. However, several of the assumptions the method relies upon have been validated in other real-world studies.

In this section the outputs of the simulation were compared across different layouts using paired confidence intervals. For the example layouts, Layout 1 and 3 are both shown to generate similarly long walking paths. Using more samples, a narrower confidence interval could be obtained, potentially differentiating one from the other. The Euclidean distance was shown to have different results than the A* path distance, which indicates it is not always an accurate representation for walking distance. Finally, a timing analysis was completed showing the speed of computing the A* path is fairly slow for the 500x500 pixel images, and options for future exploration on improving efficiency were presented.

## 5 CONCLUSIONS

This method applies simulation and computer vision techniques to provide a data-driven method for evaluating early-stage store layout plans. Steps are outlined for parsing an early store diagram into a computerized representation, generating simulated customers, computing their walking path and distance, and comparing the results across example layouts. This analysis was repeated on real proposed layouts from a local business, and results proved useful to the Store Development Committee.

There are many avenues for future work in this area. One area where this simulation could be improved is through tailoring the shopping list creation to the expected behavior of customers. Customer preferences are known to vary by age, region, and other factors. As more is known about prospective customers, this simulation can be changed to incorporate those changes and create more realistic scenarios. As the store moves forward in inventory planning, a more detailed store layout could be used, and other shelf allocation techniques applied. This information again increases accuracy, and also opens up the possibility of moving the evaluation metric from walking path lengths into specific information about the products customers pass. Combined with data on impulse buying this could translate into an estimate of the monetary increase in sales. Another extension of this work would be to incorporate optimization methods on top of the selected layout to make suggestions to the store designer to innovate on top of the existing recommendation. Finally, an important next step in any model that's based primarily on approximations and assumptions is to collect real data and compare the true results to the projected results. In this case collecting data won't be possible for all but one of the layouts, but it provides potential for future analysis, such as that done in Dorismond (2019) and Ozgormus and Smith (2020).

In conclusion, in early phases of store planning simulation can provide the opportunity to make data-driven decisions even when data is scarce. The method proposed here simulates customer walking paths to compare which proposed store layout is likely to generate the highest amount of impulse buying.

## REFERENCES

Bhadury, J., R. Batta, J. Dorismond, C.-C. Peng, and S. Sadhale. 2016. "Store Layout Using Location Modelling to Increase Purchases". University of Buffalo working paper. http://www.acsu.buffalo.edu/~batta/batta%20et%20al.pdf.

Campbell, J. 2020, April. "What is the Profit Margin for Grocery Stores?". *The Grocery Store Guy*. https://thegrocerystoreguy.com/what-is-the-profit-margin-for-grocery-stores/.

Dorismond, J. 2016. "Supermarket optimization: Simulation modeling and analysis of a grocery store layout". In *Proceedings of the Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 3656–3657. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Dorismond, J. 2019. *Data-Driven Models for Promoting Impulse Items in Supermarkets*. Ph. D. thesis, State University of New York at Buffalo.

Hart, P. E., N. J. Nilsson, and B. Raphael. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics* 4(2):100–107.

Hoch, S. J., and G. F. Loewenstein. 1991. "Time-Inconsistent Preferences and Consumer Self-Control". *Journal of Consumer Research* 17(4):492–507.

Hui, S., P. Fader, and E. Bradlow. 2009. "Research Note —The Traveling Salesman Goes Shopping: The Systematic Deviations of Grocery Paths from TSP Optimality". *Marketing Science* 28(3):566–572.

Instacart 2017. "The Instacart Online Grocery Shopping Dataset 2017". http://www.instacart.com/datasets/grocery-shopping-2017.

Kollat, D. T., and R. P. Willett. 1967. "Customer Impulse Purchasing Behavior". *Journal of Marketing Research* 4(1):21–31.

Larson, J. S., E. T. Bradlow, and P. S. Fader. 2005. "An exploratory look at supermarket shopping paths". *International Journal of Research in Marketing* 22(4):395–414.

Ozgormus, E., and A. E. Smith. 2020. "A data-driven approach to grocery store block layout". *Computers & Industrial Engineering* 139:105562.

Psparks 2017, Nov. "Instacart Market Basket Analysis". http://www.kaggle.com/psparks/instacart-market-basket-analysis.

University of Southern California 2020, Mar. "Psychology of the Grocery Store: USC Online". *USC MAPP Online*. https://appliedpsychologydegree.usc.edu/blog/psychology-of-the-grocery-store/.

## AUTHOR BIOGRAPHIES

**KIMBERLY HOLMGREN** is a Master's student in Analytics at the Georgia Institute of Technology as well as an applied researcher at the MIT Lincoln Laboratory focusing in Humanitarian Assistance and Disaster Relief Systems. She has a Bachelor's degree in computer science from the University of Chicago. Her research interests include predictive analytics, machine learning, seasonality modeling, and decision making under uncertainty. Her email address is kholmgren3@gatech.edu.