Flash Sale - Summary

Flash sales are often held when suppliers have surplus stock which they need to clear. This will attract many customers from all walks of life, trying to get themselves a good bargain. Due to the limited supply of goods available and extremely high demand, there will a be a shortage of goods. Therefore, customers will have to rush towards the shelves in order to obtain their desired product before other customers, leading to stampedes as customers jostle and push around to get ahead of others. Customers may knock products off shelves or bump against them, causing damage to other products. In this study, we will investigate how products should be placed in a fixed given floor plan to minimise damage to surrounding goods. In addition, we will also determine the optimal floor plan to reduce the likelihood of contact between customers and goods.

We developed a function to calculate how desirable each good is based on the customer rating and percentage change in price. We developed a mathematical model to calculate the destructiveness of a path taken from the entrance to reach every product, and hence calculated the total destructiveness based on a given store layout, with each *unit square* of space on a shelf containing only one type of product. We based this function on a variety of factors, with the main factor in our model being the desirability of the product, which gives a good indication of the demand for the product. Other factors included are the probability of damage (i.e. ease of damage) of a product, the flash sale price of the product as well as the width of the paths leading to the good. Based on this function, we used the *Genetic Algorithm* to shortlist the best possible arrangements of goods after designating the general location of each department, which is displayed in Figure A. This arrangement would yield the lowest destructiveness to all goods, implying minimal losses in revenue.

Based on our mathematical model for the total expected destructiveness of a store arrangement, we determined the optimal configuration of shelves. We introduced the concept of semicircular shelves to display products. The unique shape of the semicircle due to its curvature allows for space efficiency, while reducing total damage to goods on the shelf due to minimal contact with the shelf, even in the worse case scenario. We also introduced multiple wide main walkways to improve the flow of customers, hence greatly reducing congestion, and reducing damage by not placing products on main walkways where there is high human traffic. Our proposed floor plan yielded a much lower total destructiveness value at only 16.5% of the original value obtained from the fixed floor plan provided.

In our sensitivity analysis, we experimented with varying values of our coefficients in our model to determine the amount of expected destructiveness in the store, and we found that our model works perfectly with these different values and we obtained similar results with our *Genetic Algorithm*.

We have summarised the above into a letter to the Store Manager to point out important measures to be taken when organising a flash sale, propose an optimal arrangement of items based on the provided layout, as well as suggest a new store configuration (i.e. floor plan) to help the Store Manager reduce damage to products in the store and maximise profits. We believe that this is the optimal model to determine store layouts and floor plans for flash sales.

Table of contents

I Letter to Store Manager

To: Store Manager

From: Team XX

Subject: Suggestions to minimise damage to your products in the flash sale

Dear Store Manager,

Attached is our proposed floor plan layout that will allow you to make the most out of your flash sale, by minimising damage to the products on the shelves and thus minimising losses. In this plan, we have included novel semicircle shelves, which will minimise damage to products on the shelves. This is because customers do not need to walk by other products to access their desired product, and instead can head straight for the product that they wish to obtain, avoiding other products. These shelves will have an outer diameter of 9.6 m and a width of 1.6 m to maximise shelf space. On the contrary, in a grid pattern with rectangular shelves, customers would have to pass by many other products on adjacent shelves to reach their desired items, potentially causing damage to products they walk by, especially in chaotic situations.

However, due to the large number of products on sale, it is spatially impossible to fit all your items on semicircular shelves while leaving sufficient walking space. Hence, along with the semicircle shelves, we have decided to use a grid layout in the middle with three main paths, one down the middle and two on the sides, as well as 6 secondary paths on each side of the middle path. We were able to place 14 rectangular shelves in total on the two sides of the middle path, each with dimensions 1.6 m by 14.4 m and having two sides, and hence the items are accessible from the secondary paths but not the main paths, preventing damage to products due to the high human traffic on the main paths. The maximum distance a customer needs to travel off the main paths to obtain the desired product is only 7.2 m, half of the length of a shelf, decreasing likelihood of damaging other products on the way, hence preventing losses.

We believe that it is extremely important that the customers have near perfect information at any point of time to prevent unnecessary travelling and hence unnecessary damages products, and thus you should release the floor plan online ahead of time, as well as have physical copies of it at the entrance of the store so customers can head directly to where their desired good is, minimising distance travelled and hence damage to other goods. Do also make sure that there are sufficient cashiers, to prevent a long line forming and potentially extending into the main paths, increasing congestion and potentially causing damage to products. We would also suggest that you limit the number of people that can enter the flash sale at once, as a lower number of people would significantly reduce pushing and squeezing, thus protecting your products from unnecessary damage.

Thank you for giving us a chance to help out in the preparation, and we wish the flash sale a great success!

Regards,
Team XX

II Store layout for task 2(c)

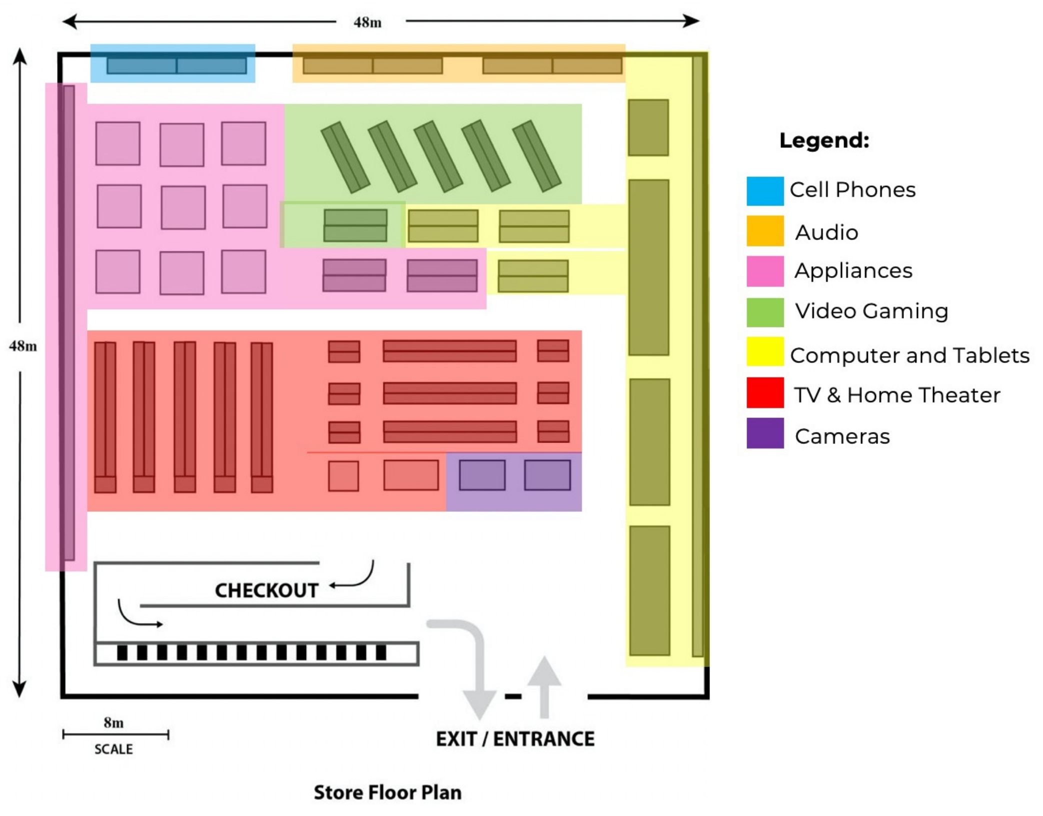


Figure A, Allocation of Departments

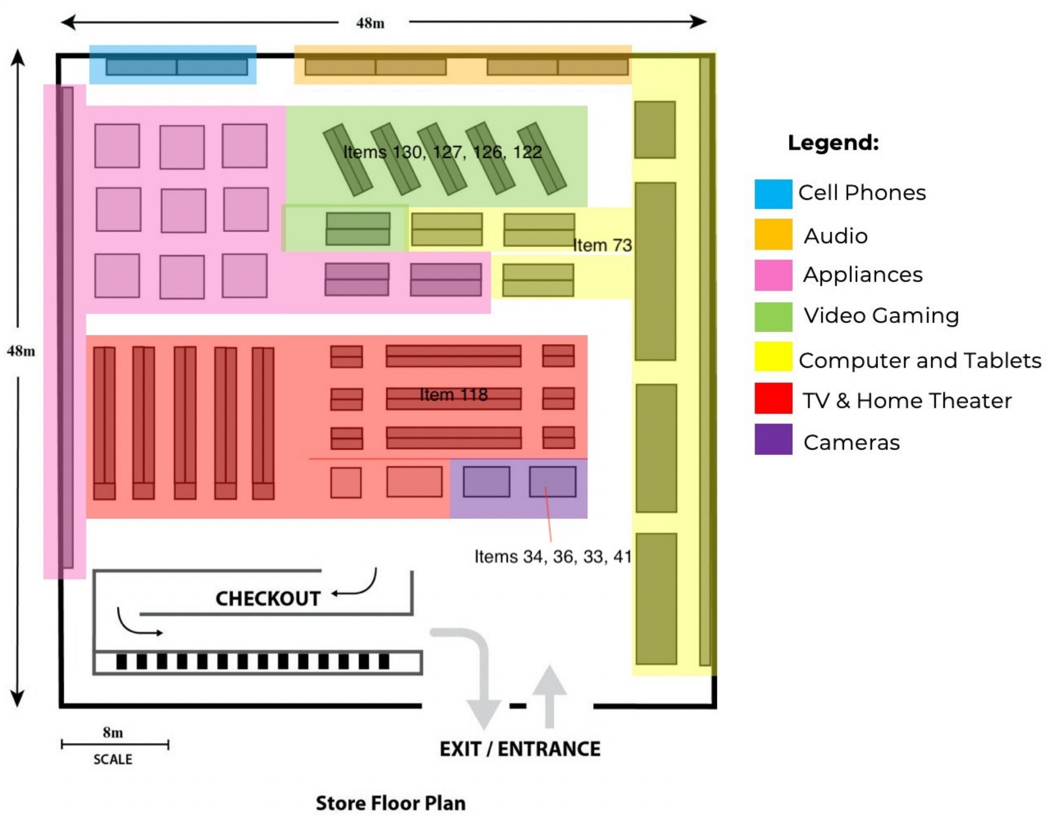


Figure B, Labelling of Top 10 Goods

*Note: Item numbers are defined in section 7.1.

III Store Layout for Task 2(d)

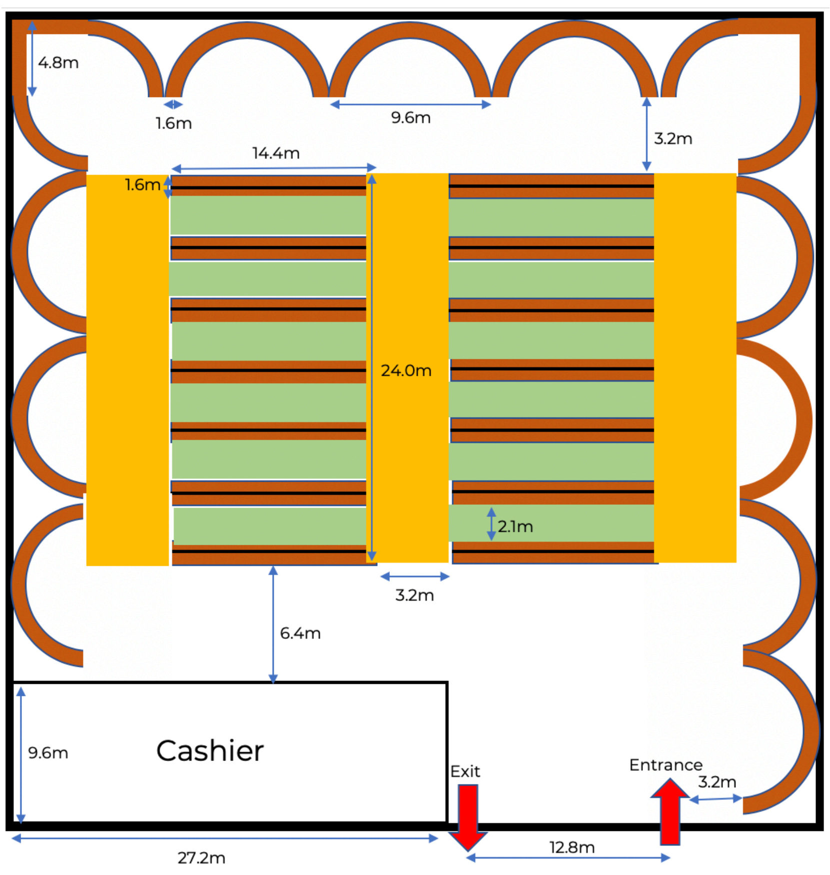


Figure C, Optimal Floor Plan with dimensions

*Note, the raw value for 2.1 m in figure C should be $\frac{32}{15}$ m. 2.1 m is rounded to 1 decimal place.

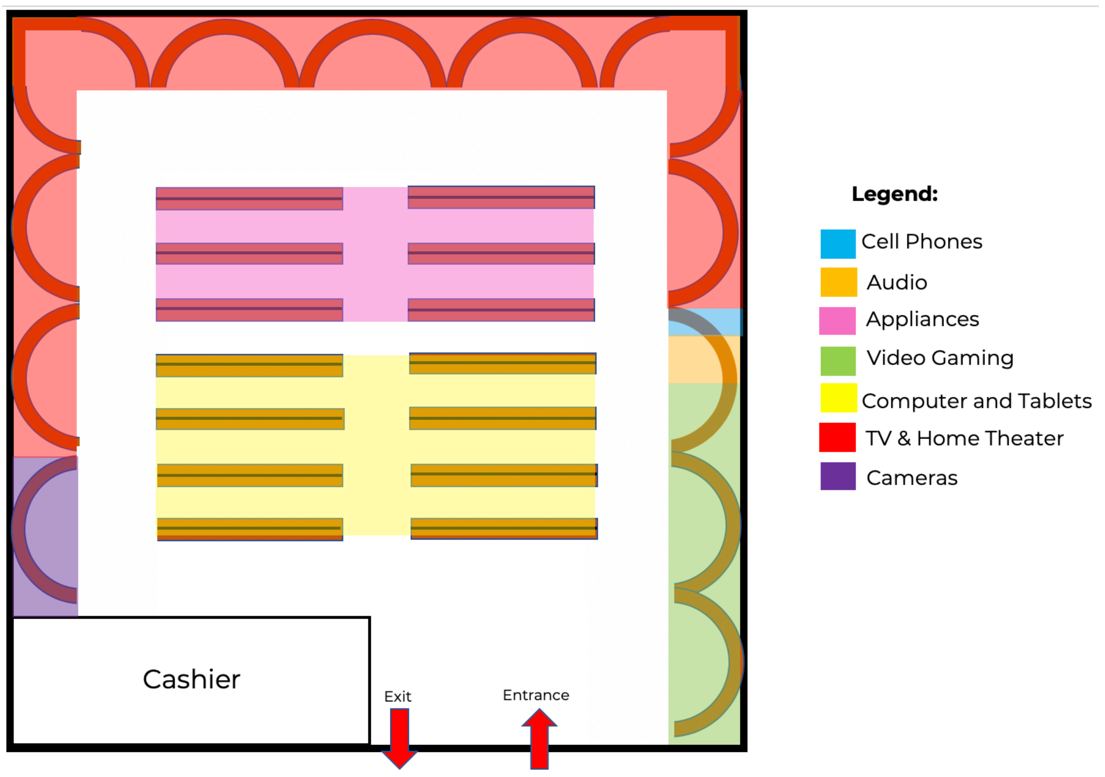


Figure D, Allocation of Departments for Optimal Floor Plan

1 Introduction

1.1 Restatement of Problem

Flash sales are common occurrences across the globe, where everyday goods are sold at a price much lower than market price in hopes of attracting customers, allowing for companies to increase their sales and clear surplus stocks. Many customers flock to flash sales in hopes of purchasing their desired goods at a bargain. However, due to the sheer number of people at flash sales, it is inevitable that there will be products damaged by eager customers who rush to their desired product and put other products in harm's way. Damaged products can no longer be sold and hence retailers incur losses. Therefore, it is paramount that stores have the most efficient layout to reduce stampedes and the number of goods damaged along the way. In essence, the problem requires us to complete the following tasks:

1. Identify how products can be damaged by customers
2. Identify which products are most sought after by customers
3. Identify factors of the store layout that would minimise the damage to products
4. Develop a model to predict how customers will navigate the store to their desired products to predict the amount of damage caused by customers in the flash sale
5. Determine the optimal locations of the different products on the provided floor plan
6. Create an efficient store layout to minimise damage to the goods in the store

1.2 Question Analysis

The most important variables when determining how products should be placed, or an efficient store layout, would be the width of the walkways and the arrangement of products according to their desirability (or popularity), in order to minimise damage to products in the store, hence minimising losses for the company. In this study, we would determine the destructiveness level of a layout based on the sum of destructiveness across all paths taken by customers to obtain the products they desire. Using this function, we would use Genetic Algorithm to repeatedly select optimal combinations of product allocation in shelves to minimise damage to products and revenue losses for retailers. In addition, we would predict the optimal store floor plan (i.e. arrangement of shelves) so that customers will cause the least amount of damage to surrounding goods using optimal semicircular shelves to minimise damage to surrounding products.

2 Assumptions and Variables

2.1 Assumptions

1. Customers have access to the floor plan, so they know the exact location of each good.

   Justification: It is extremely easy to distribute the floor plan through either online or physical methods. The floor plan can be made available via posting it online or having a physical copy displayed outside the store.

2. Customers know exactly what they want to purchase before the flash sale.

   Justification: In a flash sale, people are likely to know exactly what they intend to purchase instead of deciding on the spot. This is because the decrease in price would cause the quantity demanded to be much higher than the quantity supplied, resulting in a shortage. Hence, the goods are likely to be cleared out quickly, and taking a long time to decide would cause the customer to be unable to purchase their desired good. Hence, customers are likely to know exactly what they are intending to purchase before the flash sale commences.

3. Customers are able to afford the product they desire.

   Justification: The reason why customers come to a flash sale is to buy all their desired goods at an extremely low price. Since the product is more affordable to them, it is reasonable to assume that customers will be able to afford the goods that they desire.

4. Customers will pick one of the shortest and least congested paths to reach their desired product, which is most optimal.

   Justification: Customers want to reach the product in the shortest amount of time before the product runs out, hence they would choose the least congested and shortest path in the grid.

5. Each customer has equal probability of destroying any goods along their chosen path.

   Justification: Accidents are always bound to happen and are unavoidable in many cases. By trying to obtain the good they desire, every customer has an equal chance of bumping into or knocking over goods along the way unintentionally.

6. Every customer will not be wary of his/her surroundings to avoid hitting into products if it means slowing him/her down.

   Justification: Customers are only concerned about their own self interest as they want to get to the product they desire as quickly as possible, hence they will not be extra careful not to damage surrounding products which they would not be buying. Therefore it is inevitable that there would be damage to products on the path of the customers.

7. All products in the flash sale would be sold out eventually.

   Justification: Products in a flash sale are usually in high demand and have attractive discounts. Hence it is extremely unlikely that there will be any stock leftover after the flash sale.

8. Customers only pick one product to buy in the flash sale.

   Justification: This is because flash sales occur over a very short period of time. Products often run out of stocks very fast (as observed in the quantity of each product being at most 25) and there is likely not enough time to purchase multiple goods. Customers often come prepared with research to buy the product they have been wanting for a long time, so that product would be of the utmost priority. In the unlikely scenario that the customer really wants to purchase multiple goods, it is likely that they will get friends or family to tag along to be able to reach the product before others so that they can buy the product before stocks run dry. This would still contribute to the total number of people in the store. Hence it is extremely unlikely that one customer is able to be lucky enough to purchase so many goods that our assumption becomes invalid.

   In addition, even if customers buy multiple goods per person, they would likely visit the same department for the goods they want to buy in order to stand a chance to grab the products before they run out, hence the products purchased are very likely to be in same location and will cause negligible additional damage to other products in the store.

9. Customers will cause negligible damage to products in the store after obtaining the product they wish to purchase.

   Justification: Customers will only be rushing to obtain the product as there might be a possibility that they will not reach the product in time before other customers who wish to buy the same product. However, once they have picked the product (i.e. removed the product from the shelf), they would no longer be in a hurry to get to the cashier as they would already have their product and there would be no need to rush. Hence damage caused after they have reached their product will be minimal and insignificant.

2.2 Definition of Variables

2.2.1 Common variables

Variable	Definition
P_i	Final price (after discount) of a specific product *i*
C_i	Percentage decrease in price of a specific product *i*
R_i	Customer rating of a specific product *i*
D_i	Desirability of a specific product *i* during the flash sale given that a customer wants to buy the product in the first place. A higher value of D_i would mean that the product is more desirable. This indicates that more customers will be willing to purchase the product.
$\lambda_{S,E}$	The magnitude of the expected destruction of products from taking all possible shortest paths from vertices *S* to *E* (as defined on the store layout).
d_i	Probability of inflicting damage to specific product *i* given that customers handle the product irresponsibly.
$\mu_{S,E}$	Number of people passing through the path from vertices *S* to *E* (as defined on the store layout)
$W_{S,E}$	Width of path from vertex *S* to *E* (as defined on the store layout)
S_i	Quantity of specific product *i* of (with its packaging or boxing) that can be fit into a shelf of base area 0.8 m by 0.8 m.
F_i	A scale from 1 to 5, indicating proportion of fragile components in a specific product *i*
M_i	Mass estimate of a specific product *i* based on values found online (refer to 7.1)
n_i	Quantity available for specific product *i* during the flash sale

Table 1. Definition of variables

2.2.2 Definition of *unit square* and *square unit*

We define a *unit square* as the area taken by a square of length 0.8 m. This measurement was chosen due to the size of the shelves, all of which have dimensions which are multiples of 0.8 based on the provided floor plan. Hence, our choice of the size of the *unit square* will allow the capacity of the shelves to be expressed in terms of how many *unit squares* they cover.

A *square unit* is defined to be 0.8 m for ease of calculating.

3 Task 1

3.1 Cause of Product Damage

- When customers take a good off the shelf, they might scratch or accidentally knock adjacent goods off, damaging products. This is especially significant in flash sales, where the large crowds would cause knocking products off shelves more likely. Broken products not only contribute to sale losses, but also causes a safety problem to surrounding customers, and can cause even more congestion by blocking paths as the path may be narrow, potentially causing further damage to other goods in the vicinity.
- Products may also be damaged if there is a great demand for a certain product, hence there may be disputes between customers for a certain product. Hence, there may be potential conflicts which can cause damages to the good in the process. This occurs if there is a low supply but high demand, resulting in a shortage. Therefore, customers may fight over a certain good.

3.2 Determining Popularity of Products

3.2.1 Calculating D_i

The popularity of products in the flash sale can be determined by the desirability value of a certain product of a major product category (D_i), for example a camera package (major product category) that contains mirrorless camera with lens (specific product). D_i can be calculated as the following below:

$$D_i = (e^{R_i})(\ln(e + eC_i))$$

D_i can be represented as directly proportional to the exponent of R_i. A higher value of R_i is increasingly more difficult to achieve, since a rating closer to the maximum possible rating means that customers have no complaints at all about the product. However, in reality, this is untrue as customers will almost always find certain flaws with the product, hence the product is bound to be imperfect. Therefore, it would be increasingly difficult to satisfy customers wants further as the rating increases. Furthermore, it is likely that it would become increasingly expensive to eliminate such problems since perfection is near impossible to obtain. This increasing level of difficulty can be represented by the exponent function that has an increasing positive gradient function.

D_i can also be represented as directly proportional to the logarithmic value of C_i. As C_i increases, the increase in the desirability for the good decreases as customers see the increase in C_i as valuable as customers gain less satisfaction per unit increase in C_i as C_i increases. This can be represented using the natural logarithmic function since the function has a decreasing positive gradient function. The constant e in the function is to appropriately scale C_i to prevent the value of D_i from becoming negative.

The results of this task will be discussed in section 5.1.

4 Task 2

4.1 Factors of the Store Layout Affecting Damage to Products

4.1.1 Factors Affecting Damage to Products and Extent of Damage to Products

1. Width of Path between Shelves ($W_{S,E}$)

   The wider the path between shelves, the more people the path will be able to accommodate at any point of time, reducing the probability of customers knocking into or rub against products on the adjacent shelves. Hence $\lambda_{S,E}$ is inversely proportional to $W_{S,E}$.

2. Ease of Inflicting Damage to Products

   We can calculate the degree of ease of inflicting damage to products (Z_i) through considering the mass and the fragile components in the product. As the mass of the product increases, it is less likely that the product will get pushed off its position due to its higher weight. As the mass of the product increases, the decrease in likelihood of the product getting damaged per unit increase of mass will decrease, since it will take much more effort to push a product from its position as it gets heavier. Hence, it is appropriate to express d_i as a function of the logarithm of $\frac{1}{M_i}$. As the number of fragile components increases, the likelihood of the product getting damaged will increase. However, as the proportion of fragile components that make up the product increases, the increase in likelihood of the product getting damaged per unit increase in proportion of fragile components will decrease, since as the products are made up of more fragile components, an additional fragile component or a slight increase in the fragility of the components will not make much of a difference since it is almost equally likely that the one fragile component of product can get damaged. Hence, it is appropriate to express Z_i as a function of the logarithm of F_i to reflect the decrease in the rate of increase of Z_i for per unit increase of F_i. Hence, Z_i can be expressed as the following: $Z_i = \ln\left(\frac{F_i}{M_i} + e\right)$. The constant e is to prevent Z_i from becoming less than 1.

   As Z_i increases, it means the product is easier to damage. Hence, the probability of the product not getting damaged will decrease. This probability that the product will not get damaged can be expressed as a reciprocal of Z_i, since as Z_i increases, $\frac{1}{Z_i}$ decreases. Hence the probability of the product getting damaged is $d_i = 1 - \frac{1}{Z_i}$.

The values we used are shown in the appendix in section 7.1.

Do note that the values of M_i and F_i were consulted with references to a reliable online source or company which produces or sells that product which is shown in section 7.1.

3. Number of Products on Shelves along Path

   The more products on the shelves along a path, the higher the probability that a customer would knock into or scratch some of them. The likely cost of the damage incurred to products along a certain shelf is the sum of the cost of all probable damage done to the products, which can be expressed as $\sum d_i P_i$. We can infer that $\lambda_{S,E} \propto \sum d_i P_i$.

4. Number of People Traversing the Path

   The more people passing through a path at a certain point of time, the more likely people are to damage items on adjacent shelves. Considering that each good is at most 0.8 m wide, it is reasonable to assume that each person will occupy a width of 0.8 m. Since only people that are next to the shelves will damage goods from the shelves, the people at the 2 sides of the paths will occupy a total width of 0.8 m + 0.8 m = 1.6 m out of a possible width of $W_{S,E}$. Hence, the total number of people that can inflict potential damage is $\frac{1.6\mu_{S,E}}{W_{S,E}}$. Hence,

$$\lambda_{S,E} = \frac{1.6\mu_{S,E} \sum (d_i P_i)}{(W_{S,E})^2}$$

4.1.2 Important Measures During a Flash Sale Event

1. There has to be a limit on the total number of people allowed to enter the sale (i.e. the number of people in the flash sale cannot be so high that there is no empty space at all for customers to move around). Therefore, the number of customers cannot be too many to allow walking space in corridors and for goods to be transported after they are taken off the shelves to the counter. By ensuring sufficient space is provided, congestion can be greatly reduced and therefore accidental damage to the goods can be kept to a minimum.
2. Efficient information relay must be available so that customers are immediately notified if a good is out of stock, so that the goods which are out of stock will no longer contribute to congestion. If customers do not receive updates on the stock situation of the goods, they would be trying to visit the location of the good while causing destruction along the way, which would be sub-optimal.
3. Store directory must be located at an obvious location (e.g. right next to the entrance), so that all customers are able to view the directory and plan their route before the start of the flash sale.
4. Signages must be present everywhere in the store, so that users will almost never get lost and contribute to congestion in the store which can potentially lead to damages to surrounding products.

5. There must be enough manpower at the cashier (operating at maximum capacity) and an efficient cashier system so that the queue at the cashier will not extend beyond the checkout area into the corridors and contribute to congestion.
6. Goods from the same department should be placed in the same general area to allow the customers to easily find a specific good that they desire.
7. Pathways and directions must be clear cut, and there has to be adequate lighting in the store so that customers are able to navigate to their desired product easily and not stay in one place for too long and block the paths of other customers, contributing to congestion of walkways if they get lost.
8. There must be CCTV cameras located around the store so that customers do not intentionally damage goods in the store as the presence of CCTV cameras would deter people with ill intentions. Security guards should be present to break up fights which may arise due to disputes over certain popular goods, so that goods will not be damaged in the process.

4.2 Model Predicting Behaviour of Customers

4.2.1 Specific assumptions

- All customers are in a rush to get their desired product, hence all of them have equal potential to destroy any items.

  Justification: During a flash sale when prices are heavily discounted, it is very likely that customers want to be the first to grab their desired items, which causes them to rush for the item. This is the case because customers have to rush for their desired goods as fast as possible, so it is appropriate to assume that their damage to goods will be maximal or almost maximal should they knock something off the shelves at high speed.

4.2.2 Calculations

As discussed in section 4.1.1, the 2 formulae below shows how we can predict the behaviour of customers that potentially result in damage and the level of that damage.

The expected cost of the damage along a corridor/path from S to E can be found by:

$$\lambda_{S,E} = \frac{1.6\mu_{S,E}\sum(d_iP_i)}{(W_{S,E})^2}$$

where vertex (intersection) S and E are connected by path/corridor only, and probability of inflicting damage on a specific product i can be found by the following equation:

$$d_i = 1 - \frac{1}{\ln\left(\frac{F_i}{M_i} + e\right)}$$

The number of people passing through the path from S to E ($\mu_{S,E}$) can be calculated from the desirability values of the goods placed in the shelves along the path from S to E. As the desirability value of the goods increases, this means that more people will want to buy the good. It is most likely that the most popular good with the highest D_i (that is, $(D_i)_{\max}$) on the shelves along the path from S to E will attract most of the crowd. Hence, it is appropriate to say that $\mu_{S,E}$ is proportionate to $(D_i)_{\max}$. Hence, $\mu_{S,E} = (D_i)_{\max} \times k$ for a constant k. As d_i contains the constant k for all products, we can omit k since all values of d_i will be used for comparison only. Hence,

$$\lambda_{S,E} = \frac{1.6[(D_i)_{\max}] \sum (d_i P_i)}{(W_{S,E})^2}$$

4.3 Optimal Locations in Original Floor Plan

4.3.1 Definition of Vertices

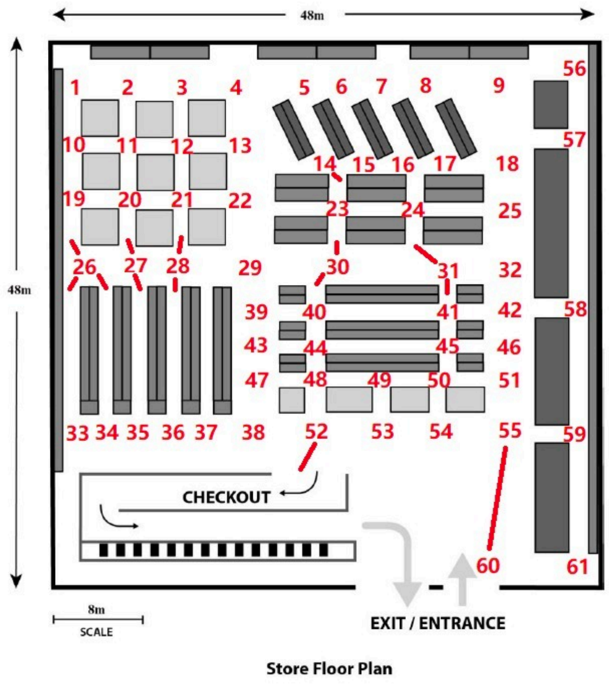


Fig. 4.3.1.1. Intersections marked out as vertices

Each vertex is defined as the intersection points between two paths (i.e. corridors) in the store, as labelled below from numbers 1 to 61 inclusive. These vertices indicate when customers can make a choice to choose a path from the intersections.

4.3.2 Unit Squares

As explained in section 2.2.2, a *unit square* is defined as the area of a square of length 0.8 m. After obtaining appropriate measurements, it was determined that the total number of *unit squares* that the shelves in the floor plan take up is 867 *unit squares* based on the floor plan. The number of *unit squares* a shelf takes up is calculated by measuring its length and width using the provided scale, then multiplying the two readings together to obtain the number of *unit squares* a shelf takes up with a reasonable uncertainty. This *unit square* system would allow us to easier judge the total area we have to place goods, allowing us to better determine the amount of each good on a shelf.

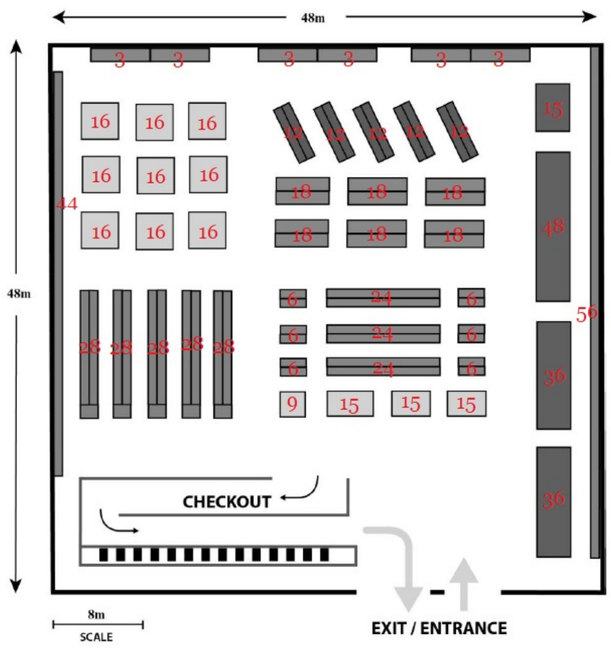


Fig. 4.3.2.1. Number of *Unit Squares* each Shelf Contains

4.3.3 Algorithm to Generate Allocation of Goods

We define $\frac{1}{x}$ to be the *mutation rate* of our Genetic Algorithm. The logic of the algorithm is as follows:

1. The shelves are first separated into seven distinct sections, one for each Department to make it easier for customers to locate their desired good.
2. The sections are each allocated a Department based on the floor space they take up.
3. Within each section, 20 random arrangements of the goods are generated. This first generation is fully randomised.
4. Total destructiveness, calculated by $\sum \lambda_{S,E}$, is calculated for each arrangement. The 10 arrangements with the highest total destruction values will be deleted.

5. The 10 remaining arrangements will *mutate*. The *mutate* process is done by creating a copy of each of the 10 remaining arrangements, picking $\frac{1}{x}$ of the items, and shuffling these items pseudo-randomly, where $\frac{1}{x}$ is the *mutation rate* further discussed below in section 4.3.4. This simulates genetic evolution in nature, where organisms mutate and evolve, with the weaker organisms dying off, allowing the organisms to become stronger over the generations. Hence, Genetic Algorithm allows the best arrangement to be generated after many generations, from purely random initial layouts.
6. Steps 3 to 5 are repeated for many generations to generate the optimal allocation for every department. The departments are combined to obtain the best allocation for the whole floor plan.

Refer to section 7.3 in the appendix for the C++ source code.

4.3.4 Discussion of mutation rate

The *mutation rate* is crucial in the Genetic Algorithm. A balance needs to be struck between *exploration,* where the algorithm experiments with a wider variety of parameters, and *exploitation* — where the algorithm makes small adjustments to the parameters to fine-tune the better solutions.

Hence, we experimented with various values of x, and ran the Genetic Algorithm for 10,000 iterations. Our results are as follows:

x	10	15	20	25	30	35	40	45
$\sum \lambda_{S,E}$	5601995	5515005	5463946	5443032	5406268	5406079	5412991	5439107

Hence, we decided the value of x to be 35.

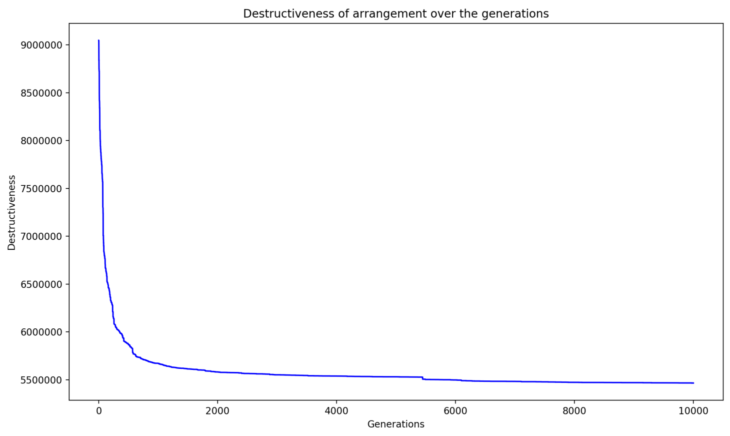
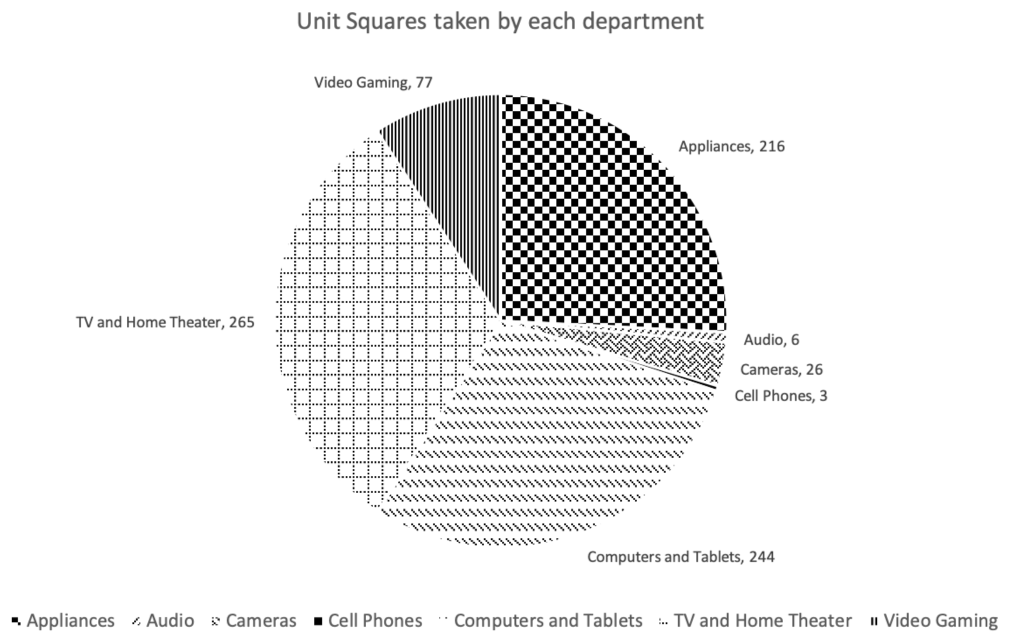


Figure 4.3.4.1. Total Destructiveness of an Arrangement over Generations

Figure 4.3.4.1 shows the total destructiveness of an arrangement over the generations. As observed, 10000 generations is sufficient to reach a convergence.

4.3.5 Allocation of Departments



4.3.5.1. Pie chart showing Division of Floor Space Occupied by each Department

Firstly, we allocate the departments in the store using the amount of floor space occupied by each department. Floor space occupied of one item $= \frac{n_i}{S_i}$

Based on these statistics, we decided to allocate the departments as seen in Figure A on page 4, so that sufficient space is allocated for each department, and each shelf contains only products from 1 department.

We used the Genetic Algorithm to allocate the products in the respective departments. We then identified where the top 10 desirable goods were located, and labelled them in Figure B. We notice that the algorithm tends to place desirable goods closer to the entrance to minimise the destructiveness, with the exception of video games. By placing desirable goods closer to the entrance, customers who rush to get these desirable goods will pass by the least number of other goods, causing less destruction to other goods.

The exception of video games is likely due to the fact that placing video games on those 5 slanted shelves will allow them to be more accessible from the wider paths, so customers will pass by a smaller number of other goods when they head towards their desired good, reducing the damage inflicted on other goods.

4.4 Creating an optimal floor plan to minimise damage to goods

The attached store floor plan on page 5, Figure C, shows the most optimal floor plan to minimise the expected loss of revenue due to the rowdy behaviour of customers during flash sales.

A shelf is a structure or open area where a product can be displayed, and its capacity to hold products (or area) is measured in *unit squares* for convenience. A rectangular shelf is defined as a rectangular block where products can be placed on both longer sides of the block, while the shorter sides of the rectangle do not display any goods, to reduce damage caused. Rectangular shelves are represented using brown rectangles as shown in the figure above. Semicircular shelves are represented by concave brown semicircles which are capable of storing products only on the inner side of the semicircle.

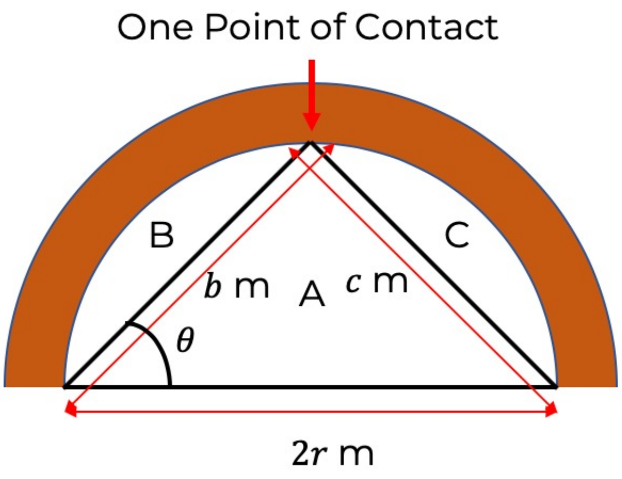


Fig 4.4.1. Diagram of a semicircular shelf, where there is only one point of contact when customers take goods from the shelf

In order to obtain the least congestion, with customers coming into least contact with other goods, we observe that shelves of semicircle shapes are the most efficient, where goods are placed on the inner surface of the semicircle such that customers will only visit a point in the inner side of the semicircle. These semicircular shelves are placed at the sides of the store. When customers visit the shelf to obtain a particular product, they will travel in a straight line, and by doing so, there will be only one point of contact with the shelf, which is the point where the product is located. Hence, this configuration would inflict little damage on surrounding goods. We used trigonometry to calculate the average distance of a customer to the closest other good, given the value of θ, as follows:

We obtain:

$$b = 2r\cos\theta$$

$$c = 2r\sin\theta$$

$$\begin{aligned}\text{dist}_{\text{ave}} &= \frac{\text{Area of } B + \text{Area of } C}{b+c}\\ &= \frac{\frac{\pi r^2}{2} - A}{2r\sin\theta + 2r\cos\theta}\\ &= r \times \frac{\pi - 2\sin 2\theta}{4(\sin\theta + \cos\theta)}\end{aligned}$$

$$\because r = 3.2, \therefore \text{dist}_{\text{ave}} = \frac{4\pi - 2\sin 2\theta}{5(\sin\theta + \cos\theta)}$$

To find the average distance of each value of θ, we have integrated dist_{ave} from 0 to $\frac{\pi}{2}$ to find the area under the graph, and then divide the obtained value by $\frac{\pi}{2}$ to obtain average distance from products for average value of θ. This would then be multiplied by 2 to obtain the width value for the semicircular shelves, to allow it be compared to the width value of the paths between rectangular shelves where the width value measures the distance between two shelves, not the distance between the customer and the products, which can at most be the width of the path divided by two. The width is computed as follows:

$$\begin{aligned}W_{\text{semicircle}} &= \int_0^{\frac{\pi}{2}} \frac{4\left(\pi - 2\sin(2x)\right)}{5\left(\sin(x) + \cos(x)\right)} \div \frac{\pi}{2} \times 2\\ &\approx 2.4 \text{ m}\end{aligned}$$

Hence,

$$\lambda_{\text{semicircles}} = \frac{1.6}{(2.4 \times \frac{1}{0.8})^2}(105.77 \times 312 + 149.22 \times 162.77 + 116.49 \times 1051 + 141.27 \times 1281 + 170.56 \times 1770) = 492241$$

However, due to space constraints, the top right and top left corners cannot be covered with semicircles, hence the items will be placed on the perimeters of the area. Visiting these shelves will incur a certain amount of damage, which depends on the following formula (the formula will be explained with Figure 4.4.2):

$$\lambda_\alpha = \frac{\frac{1}{4}(\pi)(2)[(R-2)^2 - (R-3)^2] + (R-2)^2 - (R-3)^2}{\frac{1}{4}(2)(\pi)(R-2)^2 + (R-2)^2} \times (D_i)_{\max} \times \sum (d_i P_i)$$

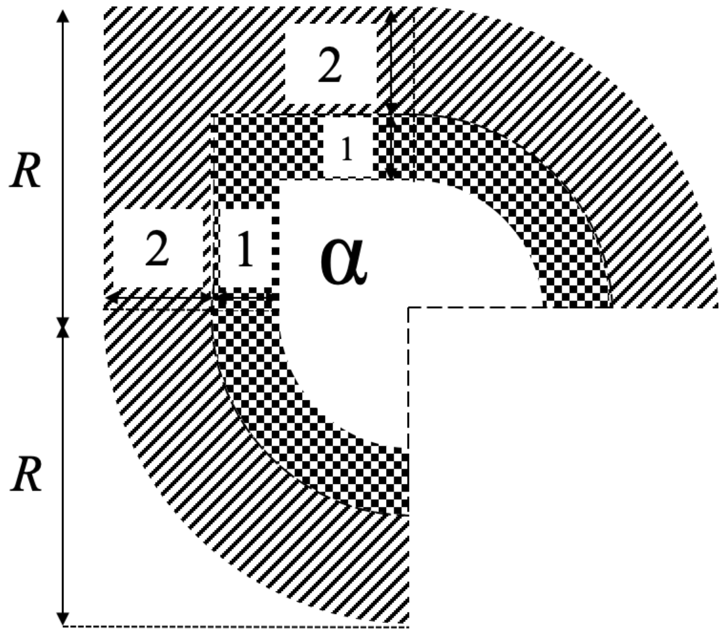


Figure 4.4.2. Dimensions for the corners (R and the constants are expressed in *square units*)

The expected revenue loss incurred from the damage λ_α in the region α as defined by the dotted lines is due to people coming into close contact with the goods placed in the striped region. At 1 *square unit* away, this is when people can inflict damage to other goods, and this is marked by the region in checkerboard. Hence, the probability that people will be in the region where they can inflict damage is the area of the checkerboard, $\frac{1}{4}(\pi)(2)[(R-2)^2-(R-3)^2]+(R-2)^2-(R-3)^2$, divided by the area of α, $\frac{1}{4}(2)(\pi)(R-2)^2+(R-2)^2$. As explained in section 4.1.1 point (3), $\lambda_\alpha \propto \sum d_iP_i$, where $\sum d_iP_i$ represents the sum of the expected damage of all goods should a customer unintentionally damage it. By multiplying by $(D_i)_{\max}$, the maximum desirability of goods in the shelves, we include the probability of damage due to desirability which will correlate to the number of people present at the location. This is multiplied by 2 as there are two corners in the plan.

$$2\lambda_\alpha = 2\times\frac{\frac{1}{2}(\pi)(2)[(6-2)^2-(6-3)^2]+(6-2)^2-(6-3)^2}{\frac{1}{4}(2)(\pi)(6-2)^2+(6-2)^2}\times 141.27\times 1281 = 158346$$

By considering the fact that there are 10 semicircular shelves, 4 quarter circle shaped shelves and 2 ‘L’ shaped shelves, each of uniform thickness 2 *square units* = 1.6 m, the outer radius of the semicircular, quarter circle and outer length of the ‘L’ shaped shelves is 6 *square units* = 4.8 m (that is R in Figure 4.4.2), the total capacity of the shelves at the perimeter of the store based on the map shown in Figure 4.4.1 is $\frac{1}{2}(3.5+4+4.5)\lfloor(6^2\pi-4^2\pi)\rfloor+2(12^2-10^2)=412$ *unit squares*.

Based on this observation, we assigned all of the products under departments of Video Games, TV, Audio, Cellphones and Cameras to the semicircular shelves, each with a width of 2 *square units*. These products take up a total of 377 *unit squares* < 412 *unit squares*, which fits the requirements with space to spare. Video games and cameras are placed closer to the entrance due to their high desirability value. Refer to Figure D for the allocation of the departments in the store.

This leaves 2 more departments: Appliances and Laptop & Tablets, which take up a total of 460 square units. Therefore, we can classify them into rectangular shelves of 36 *unit squares* each, with 18 *unit squares* on each side of the shelf, in the center of the store as shown in Figure C. There are 3 main walkways, each 4 *square units* wide (3.2 m) as indicated by the yellow region. The length of the a shelf is determined to be 18 *square units* as this is the maximum shelf allowance available to allow 4 *square units* walkways at the sides of the store, and one main *4 square units* walkway in the middle of the store. Laptops and tablets are placed closer to the entrance as they have a higher desirability rating as compared to appliances.

Traversing on these main walkways (in yellow) will not cause any damage to surrounding goods as no goods are placed on the sides of the rectangular shelves along the main pathways. Damage is only incurred when customers enter the secondary walkways (in green) from the main walkways (in yellow), when they pass by other products. Hence, the maximum possible damage in the worst case scenario would be the case when the customer traverses the entire length of the shelf to reach their product. The width of the secondary walkways (in green) would be based on the total number of *unit squares* of items to be allocated. In each vertical column, the total number of *unit squares* needed would be $\frac{460}{2} = 230$. Each shelf has capacity of 36 *unit squares*. Hence the number of shelves required for each column of shelves (which corresponds to the total number of rows) is $\lceil \frac{230}{36} \rceil = 7$. The ceiling function is required as even if part of a shelf is used, a new shelf is required. Therefore there are 7 rows in total. The total height of the middle section containing the shelves would be $60 - 6 - 6 - 6 - 12 = 30$ *square units*. Hence the width of secondary paths (in green) is $\frac{30 - 2(7)}{6} = 2.7$ *square units* (to 1 decimal place). Hence the destructiveness of the path would be the $\lambda_{S,E} = \frac{1.6[(D_i)_{\max}]\sum(d_iP_i)}{2.7^2}$. The total destructiveness of all paths across all departments placed in rectangular shelves is calculated to be 201922.

Consequently, the total destructiveness to the store, $\lambda_{\text{total}} = 492241 + 201922 + 158346 = 852509$. This is found to be much smaller than the original floor plan using Genetic Algorithm, which is 5306268. This shows that our new layout is more optimal as it will cause much less damage and revenue loss to the store by arranging and choosing furniture so that there will be the least contact between customers and goods.

5 Discussion of results

5.1 Most Popular Products

Item ID	Product Name	Department	C_i	R_i	D_i
130	15.6" Gaming Laptop, AMD Ryzen 5, 8GB Ram, NVIDIA GeForce GTX 1050, 25	Video Gaming	0.4375	5.0	170.56
127	Gamer Supreme Liquid Cool Gaming Desktop, AMD Ryzen 7 3700X	Video Gaming	0.3000	5.0	163.95
126	Gamer Master Gaming Desktop, AMD Ryzen 5 3600, 8GB Memory	Video Gaming	0.2683	5.0	162.38
34	DSLR Camera, Body Only, Black	Cameras	0.3333	4.9	149.82
73	27" IPS LED FHD FreeSync Monitor, 27f	Computers and Tablets	0.5600	4.8	144.27

Table 5.1.1. List of top 5 popular products

5.2 Sensitivity Analysis

We varied the coefficient of 1.6 in the λ function, from 1.5 to 1.7, to test if our function is consistent. We observed that our function generated a range of values of expected destructiveness between 5151345 and 5833369. These values are all quite consistent, and are still much greater than the expected destructiveness of our proposed floor plan, which only generates a destructiveness value of 852509, which is around 16.5% of the original destructiveness, making our new version always much more efficient and damage resistant as compared to the original floor plan, despite varying our coefficients.

In addition, as the *Genetic Algorithm* is generated based on a random permutation during the first generation (i.e. before the algorithm *mutates*), there is a certain element of randomness. However, this does not affect our overall results as the *Genetic Algorithm* will eventually mutate to the optimal permutation of products. After executing the program for 5 times in a row, we obtain a range of values from 5477425 - 5504264, with a total range of 26839, which is relatively insignificant as it is less than 0.5% of 5477425.

5.3 Comparison of results with other scenarios

The table below shows the summary of λ values obtained from section 4.3 and 4.4.

No.	How λ is obtained	Value of λ
1	λ_1 in section 4.3.4 by using genetic algorithm to solve task 2(c)	5.406079×10^6
2	λ_2 in section 4.4 from the more optimal store plan generated for task 2(d)	8.52509×10^5
3	λ_3 by assuming the worst case scenario if the store layout in 2(c) were to be used	1.4047108×10^7

λ_3 is obtained by reversing the *Genetic Algorithm* (in section 4.3.3), i.e. Selecting the worst, rather than the best permutations at each iteration.

Considering that λ_2 is 6.07% of λ_3, this means that at most 6.07% of magnitude of damage will be incurred should the more optimal store plan be used as compared to in the worst case scenario (i.e. worst placement of goods) if the original store plan were to be used. This shows that the proposed store plan is very efficient should goods be placed optimally. Since the original store layout proposed may not be the worst possible, this means that our proposed floor plan will only cause at most 6.07% of magnitude of damage. This greatly minimises the revenue loss for the company since there is a more than 90% in reduction in the magnitude of damage that can be potentially inflicted.

5.4 Strengths

1. The function to calculate the total expected destructiveness is an accurate description of how likely goods would be destroyed in the store. This model takes into account multiple important factors which affect the expected damage to the products in the store by the customers, such as the price of the product, the percentage change in price, the desirability, the probability of damage, and the width of the path.
2. Total destructiveness is calculated with a statistical method using *Graph Theory*. During the process, recursion and backtracking techniques were utilised, which enhanced the speed and reliability of the function to calculate total destructiveness due to its high objectivity. By using a C++ program, we were able to simulate actual customers buying the products off the shelf, using a substantial amount of data provided (1535 products, 137 unique products), which would have been difficult to compute without the aid of *Graph Theory* and *Genetic Algorithm*.
3. When considering the mass, we researched on the actual mass of each of the products to increase the accuracy and reliability of the model by using real information from suppliers and wikis

regarding these products. In addition, we also measured the dimensions of the floor plan provided to scale to determine *unit squares*, which also helped to increase the reliability of the model to ensure that all products can fit into the *unit squares*.

4. We effectively used *Genetic Algorithm* to form an arrangement of the goods in the store that minimises the total destructiveness with a fixed given layout, objectively forming ideal arrangements as seen in the convergence of the algorithm (in Figure 4.3.4.1).
5. The new floor plan that we generate causes much less destruction, with $\lambda_{\text{total}} = 852509$, 16.5% of the ideal situation with the original floor plan. This implies that a huge proportion of revenue which would have been lost in the original floor plan will be earned by the retailer as profits instead, reducing losses greatly.
6. As shelves in the floor plan are more regular, this makes it easier for customers to locate products that they desire. This will help to reduce congestion caused by lost or confused customers, therefore reducing the amount of damage caused to the products on sale.

5.5 Limitations

1. Each customer may want to take multiple goods, and this is not accounted for in our model as we assume each consumer only takes a single product. This assumption may not be always true in the real world scenario. However, as mentioned above, most customers will only be able to pick one or two goods, and stay in the same department when purchasing products, hence it is very unlikely for these customers to cause more damage. For more details, refer to Section 2.1.
2. Customers may not know exactly what they want to purchase before the flash sale — they may be there for leisure shopping. This will cause our assumption to be less valid. However, even if this occurs, most customers will be more interested in a specific type of good. Therefore, they will only go to one location, and our premise will still hold.
3. Departments are assigned semi-automatically, so the allocation of department locations may not be optimal. This is due to the varying shapes and capacities of the shelves, posing significant difficulties in organising the goods. In addition, the goods come in varying shapes and sizes, which limits the flexibility of the arrangements. The departments are assigned based on the floor space all the items in each department takes up, which is computed by the formula n_iS_i. For more details, refer to Section 4.3.3.
4. We assumed that customers will take one of the shortest or least congested paths to obtain their good. While this is often the case, in reality, customers may take a slightly longer path. This will cause slightly more damage than what we predict. However, this slight additional damage will be

negligible as customers will want to reach their destination in the shortest period of time, even if they take a longer path, the distance travelled is unlikely to be much further than predicted.

6 Conclusion

To determine the optimal arrangement of products on a provided floor plan, we computed a desirability value of each product to account for the popularity of each product based on price and customer rating. This gives a good indication of the number of people who are likely to attempt to obtain the good during the flash sale. Using this function, we calculated the expected destructiveness of a given store layout and floor plan based on the desirability values of each good, as well as the price level which indicates the value of the product, and the probability of damage of a certain product as some products in the given list are damaged more easily than others due to the presence of fragile components.

With the function λ, we were able to determine how much destruction a particular layout would give, and we aimed to minimise this value when we arranged the products on the given floor plan on the shelves, using Genetic Algorithm to selectively extract the best permutations of the products on the shelves for the lowest expected destructiveness in the store, which proved to be successful as popular goods such as video games were placed near the main paths in the layout for easier accessibility and so that buyers of these popular products are unlikely to venture deeper into the store for their product, hence causing less damage to other goods, which will increase the store's profits.

Based on our function λ, we could determine the optimal floor plan of the store, by arranging semicircular shelves at the sides of the store to maximise space efficiency while at the same time reducing damage to surrounding goods greatly. In the central section of the store, we installed multiple wide main pathways for customers to easily navigate to their desired goods while passing by the least number of other goods, causing the least damage to the store, hence maximising profits for the retailer.

Future studies could involve using more actual data from flash sales to test whether this model is a suitable tool to determine the optimal layout and floor plan of a flash sale. In addition, if there was access to real data regarding the actual revenue lost due to damage to products in a flash sale, we would be able to determine a more accurate function based on more information, using methods such as regression techniques and machine learning.

7 Appendix

7.1 Data

7.1.1 Values of M_i and F_i

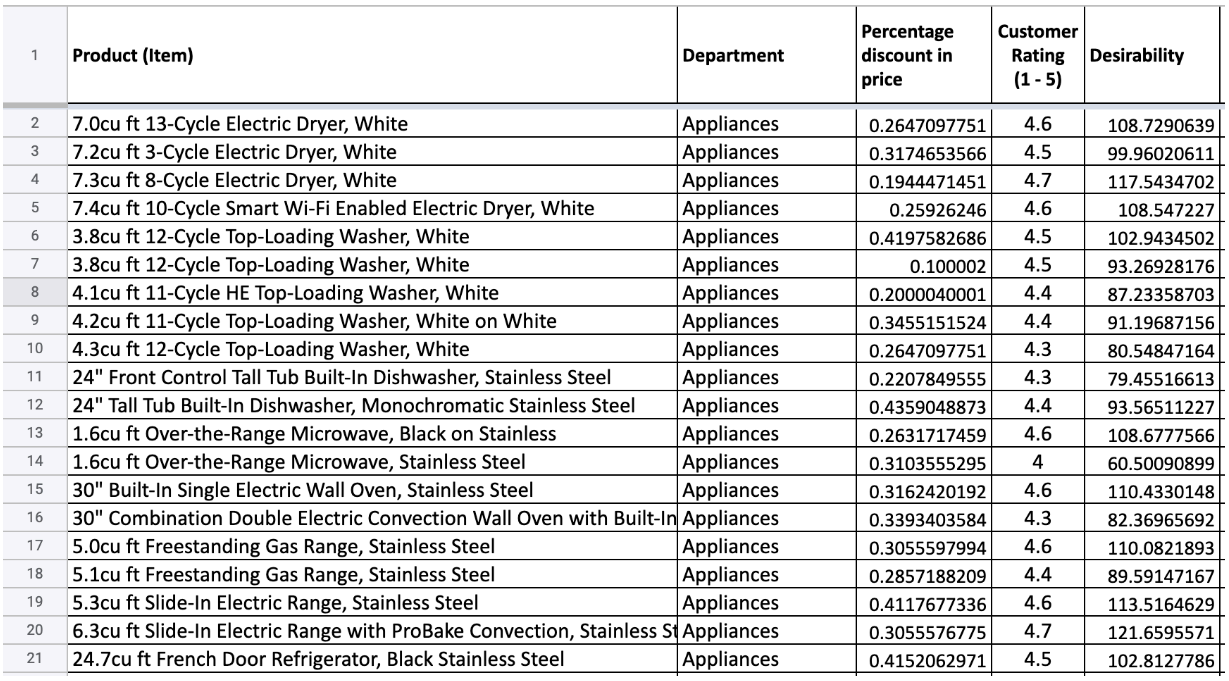
The table below shows the mass (in kg) and fragility (on a scale of 1-5) of a typical product in each Major Product Category. These are later used for computation of d_i. All data came from reliable online websites or official company websites.

Major Product Category	M_i	F_i	Source
Cell Phones and Accessories	0.2	4	https://www.pocket-lint.com/phones/news/117972-how-heavy-latest-smartphone-are
Console Game Systems	3	4	https://www.cnet.com/news/ps4-slim-ps4-pro-xbox-one-s-scorpio-spec-showdown/
Desktops and All-In-Ones	30	3	https://gamefaqs.gamespot.com/boards/916373-pc/75101257
DSLR Cameras	0.5	5	http://benmlee.com/Digital_Camera/Lens_Weight_Compare.htm
Headphones	0.3	2	https://www.headphonezone.in/pages/headphone-weight
Laptops	2	4	https://computingspot.com/how-much-does-a-laptop-weigh/
Laundry Appliances	90	2	https://www.hunker.com/13410576/the-average-weight-of-a-washing-machine
Major Kitchen Appliances	60	1	https://prudentreviews.com/refrigerator-weight/
Mirrorless Cameras	0.4	5	http://benmlee.com/Digital_Camera/Lens_Weight_Compare.htm
Monitors	6	4	https://www.cu.edu/sites/default/files/Dell_P2214H_Monitor_Spec_Sheet.pdf
PC Gaming	8	4	https://www.dell.com/en-sg/shop/desktop-computers/dell-g5-gaming-desktop/spd/g-series-5090-desktop
Printers	6	2	https://support.hp.com/ca-en/document/c00312638
Tablets	0.3	4	https://www.quora.com/How-much-does-a-55-inch-smart-TV-weigh-in-kilograms
TVS 30" to 45"	10	4	https://www.samsung.com/us/video/tvs/UN40D5003BFXZA-specs
TVS 50" - 55"	20	4	https://www.quora.com/How-much-does-a-55-inch-smart-TV-weigh-in-kilograms

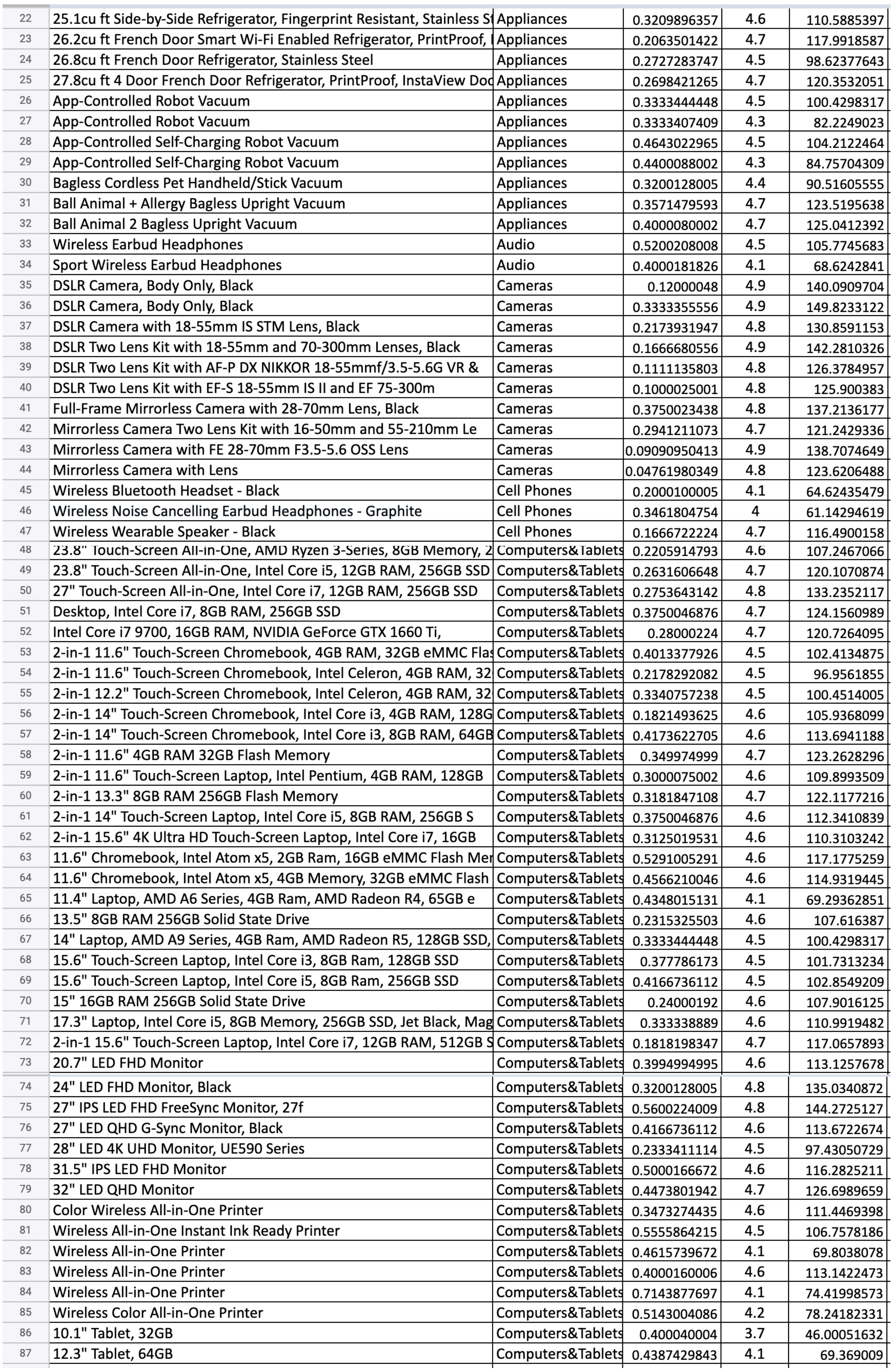
Major Product Category	M_i	F_i	**Source**
TVS 65"	30	4	https://www.bestbuy.com/site/questions/samsung-65-class-led-nu6070-series-2160p-smart-4k-uhd-tv-with-hdr/6290160/question/63c0f4c1-4bc2-31c7-87d2-704837e7f2c8
TVS 70" - 75"	40	4	https://www.amazon.com/Sony-KD70X690E-70-Inch-Ultra-Smart/dp/B072FRV68G
TVS 85"	80	4	https://www.amazon.com/Sony-XBR85X850D-85-Inch-Ultra-Smart/dp/B01A5LU5X0
Vacuum Cleaners & Floor Care	10	1	https://www.shopyourway.com/questions/1050508
Video	1	1	https://www.sony.com.sg/electronics/blu-ray-disc-players/bdp-s1500/specifications

7.1.2 Values of C_i , R_i and D_i

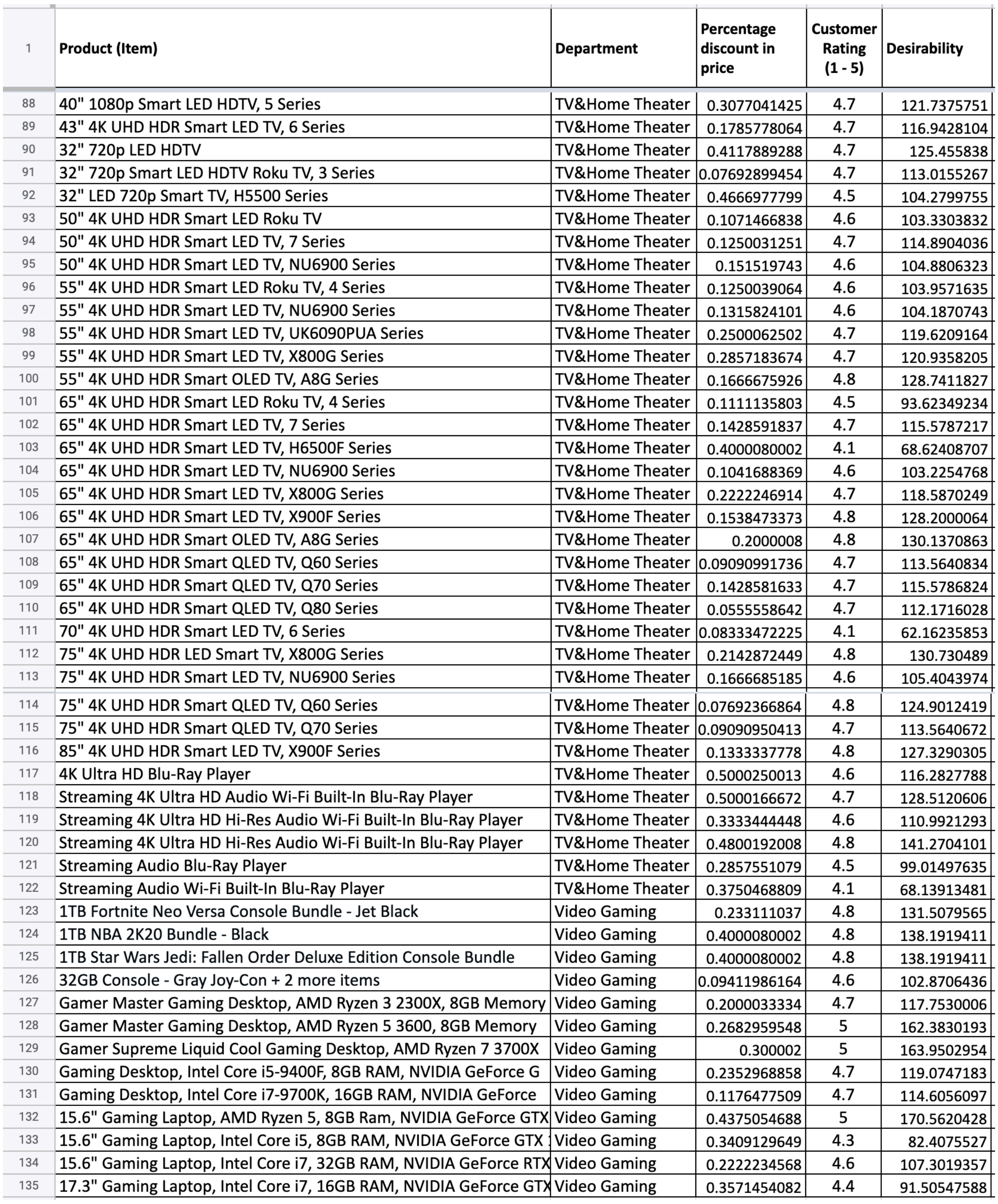
Note that item ID referenced in the above sections start from 0. (i.e. 7.0cu ft 13-Cycle Electric Dryer is Item 0; 7.2cu ft 3-Cycle Electric Dryer, White is Item 1 and so on)



1	Product (Item)	Department	Percentage discount in price	Customer Rating (1 - 5)	Desirability
2	7.0cu ft 13-Cycle Electric Dryer, White	Appliances	0.2647097751	4.6	108.7290639
3	7.2cu ft 3-Cycle Electric Dryer, White	Appliances	0.3174653566	4.5	99.96020611
4	7.3cu ft 8-Cycle Electric Dryer, White	Appliances	0.1944471451	4.7	117.5434702
5	7.4cu ft 10-Cycle Smart Wi-Fi Enabled Electric Dryer, White	Appliances	0.25926246	4.6	108.547227
6	3.8cu ft 12-Cycle Top-Loading Washer, White	Appliances	0.4197582686	4.5	102.9434502
7	3.8cu ft 12-Cycle Top-Loading Washer, White	Appliances	0.100002	4.5	93.26928176
8	4.1cu ft 11-Cycle HE Top-Loading Washer, White	Appliances	0.2000040001	4.4	87.23358703
9	4.2cu ft 11-Cycle Top-Loading Washer, White on White	Appliances	0.3455151524	4.4	91.19687156
10	4.3cu ft 12-Cycle Top-Loading Washer, White	Appliances	0.2647097751	4.3	80.54847164
11	24" Front Control Tall Tub Built-In Dishwasher, Stainless Steel	Appliances	0.2207849555	4.3	79.45516613
12	24" Tall Tub Built-In Dishwasher, Monochromatic Stainless Steel	Appliances	0.4359048873	4.4	93.56511227
13	1.6cu ft Over-the-Range Microwave, Black on Stainless	Appliances	0.2631717459	4.6	108.6777566
14	1.6cu ft Over-the-Range Microwave, Stainless Steel	Appliances	0.3103555295	4	60.50090899
15	30" Built-In Single Electric Wall Oven, Stainless Steel	Appliances	0.3162420192	4.6	110.4330148
16	30" Combination Double Electric Convection Wall Oven with Built-In	Appliances	0.3393403584	4.3	82.36965692
17	5.0cu ft Freestanding Gas Range, Stainless Steel	Appliances	0.3055597994	4.6	110.0821893
18	5.1cu ft Freestanding Gas Range, Stainless Steel	Appliances	0.2857188209	4.4	89.59147167
19	5.3cu ft Slide-In Electric Range, Stainless Steel	Appliances	0.4117677336	4.6	113.5164629
20	6.3cu ft Slide-In Electric Range with ProBake Convection, Stainless St	Appliances	0.3055576775	4.7	121.6595571
21	24.7cu ft French Door Refrigerator, Black Stainless Steel	Appliances	0.4152062971	4.5	102.8127786



22	25.1cu ft Side-by-Side Refrigerator, Fingerprint Resistant, Stainless St	Appliances	0.3209896357	4.6	110.5885397
23	26.2cu ft French Door Smart Wi-Fi Enabled Refrigerator, PrintProof, I	Appliances	0.2063501422	4.7	117.9918587
24	26.8cu ft French Door Refrigerator, Stainless Steel	Appliances	0.2727283747	4.5	98.62377643
25	27.8cu ft 4 Door French Door Refrigerator, PrintProof, InstaView Doo	Appliances	0.2698421265	4.7	120.3532051
26	App-Controlled Robot Vacuum	Appliances	0.3333444448	4.5	100.4298317
27	App-Controlled Robot Vacuum	Appliances	0.3333407409	4.3	82.2249023
28	App-Controlled Self-Charging Robot Vacuum	Appliances	0.4643022965	4.5	104.2122464
29	App-Controlled Self-Charging Robot Vacuum	Appliances	0.4400088002	4.3	84.75704309
30	Bagless Cordless Pet Handheld/Stick Vacuum	Appliances	0.3200128005	4.4	90.51605555
31	Ball Animal + Allergy Bagless Upright Vacuum	Appliances	0.3571479593	4.7	123.5195638
32	Ball Animal 2 Bagless Upright Vacuum	Appliances	0.4000080002	4.7	125.0412392
33	Wireless Earbud Headphones	Audio	0.5200208008	4.5	105.7745683
34	Sport Wireless Earbud Headphones	Audio	0.4000181826	4.1	68.6242841
35	DSLR Camera, Body Only, Black	Cameras	0.12000048	4.9	140.0909704
36	DSLR Camera, Body Only, Black	Cameras	0.3333355556	4.9	149.8233122
37	DSLR Camera with 18-55mm IS STM Lens, Black	Cameras	0.2173931947	4.8	130.8591153
38	DSLR Two Lens Kit with 18-55mm and 70-300mm Lenses, Black	Cameras	0.1666680556	4.9	142.2810326
39	DSLR Two Lens Kit with AF-P DX NIKKOR 18-55mmf/3.5-5.6G VR &	Cameras	0.1111135803	4.8	126.3784957
40	DSLR Two Lens Kit with EF-S 18-55mm IS II and EF 75-300m	Cameras	0.1000025001	4.8	125.900383
41	Full-Frame Mirrorless Camera with 28-70mm Lens, Black	Cameras	0.3750023438	4.8	137.2136177
42	Mirrorless Camera Two Lens Kit with 16-50mm and 55-210mm Le	Cameras	0.2941211073	4.7	121.2429336
43	Mirrorless Camera with FE 28-70mm F3.5-5.6 OSS Lens	Cameras	0.09090950413	4.9	138.7074649
44	Mirrorless Camera with Lens	Cameras	0.04761980349	4.8	123.6206488
45	Wireless Bluetooth Headset - Black	Cell Phones	0.2000100005	4.1	64.62435479
46	Wireless Noise Cancelling Earbud Headphones - Graphite	Cell Phones	0.3461804754	4	61.14294619
47	Wireless Wearable Speaker - Black	Cell Phones	0.1666722224	4.7	116.4900158
48	23.8" Touch-Screen All-in-One, AMD Ryzen 3-Series, 8GB Memory, 2	Computers&Tablets	0.2205914793	4.6	107.2467066
49	23.8" Touch-Screen All-in-One, Intel Core i5, 12GB RAM, 256GB SSD	Computers&Tablets	0.2631606648	4.7	120.1070874
50	27" Touch-Screen All-in-One, Intel Core i7, 12GB RAM, 256GB SSD	Computers&Tablets	0.2753643142	4.8	133.2352117
51	Desktop, Intel Core i7, 8GB RAM, 256GB SSD	Computers&Tablets	0.3750046876	4.7	124.1560989
52	Intel Core i7 9700, 16GB RAM, NVIDIA GeForce GTX 1660 Ti,	Computers&Tablets	0.28000224	4.7	120.7264095
53	2-in-1 11.6" Touch-Screen Chromebook, 4GB RAM, 32GB eMMC Flas	Computers&Tablets	0.4013377926	4.5	102.4134875
54	2-in-1 11.6" Touch-Screen Chromebook, Intel Celeron, 4GB RAM, 32	Computers&Tablets	0.2178292082	4.5	96.9561855
55	2-in-1 12.2" Touch-Screen Chromebook, Intel Celeron, 4GB RAM, 32	Computers&Tablets	0.3340757238	4.5	100.4514005
56	2-in-1 14" Touch-Screen Chromebook, Intel Core i3, 4GB RAM, 128G	Computers&Tablets	0.1821493625	4.6	105.9368099
57	2-in-1 14" Touch-Screen Chromebook, Intel Core i3, 8GB RAM, 64GB	Computers&Tablets	0.4173622705	4.6	113.6941188
58	2-in-1 11.6" 4GB RAM 32GB Flash Memory	Computers&Tablets	0.349974999	4.7	123.2628296
59	2-in-1 11.6" Touch-Screen Laptop, Intel Pentium, 4GB RAM, 128GB	Computers&Tablets	0.3000075002	4.6	109.8993509
60	2-in-1 13.3" 8GB RAM 256GB Flash Memory	Computers&Tablets	0.3181847108	4.7	122.1177216
61	2-in-1 14" Touch-Screen Laptop, Intel Core i5, 8GB RAM, 256GB S	Computers&Tablets	0.3750046876	4.6	112.3410839
62	2-in-1 15.6" 4K Ultra HD Touch-Screen Laptop, Intel Core i7, 16GB	Computers&Tablets	0.3125019531	4.6	110.3103242
63	11.6" Chromebook, Intel Atom x5, 2GB Ram, 16GB eMMC Flash Mer	Computers&Tablets	0.5291005291	4.6	117.1775259
64	11.6" Chromebook, Intel Atom x5, 4GB Memory, 32GB eMMC Flash	Computers&Tablets	0.4566210046	4.6	114.9319445
65	11.4" Laptop, AMD A6 Series, 4GB Ram, AMD Radeon R4, 65GB e	Computers&Tablets	0.4348015131	4.1	69.29362851
66	13.5" 8GB RAM 256GB Solid State Drive	Computers&Tablets	0.2315325503	4.6	107.616387
67	14" Laptop, AMD A9 Series, 4GB Ram, AMD Radeon R5, 128GB SSD,	Computers&Tablets	0.3333444448	4.5	100.4298317
68	15.6" Touch-Screen Laptop, Intel Core i3, 8GB Ram, 128GB SSD	Computers&Tablets	0.377786173	4.5	101.7313234
69	15.6" Touch-Screen Laptop, Intel Core i5, 8GB Ram, 256GB SSD	Computers&Tablets	0.4166736112	4.5	102.8549209
70	15" 16GB RAM 256GB Solid State Drive	Computers&Tablets	0.24000192	4.6	107.9016125
71	17.3" Laptop, Intel Core i5, 8GB Memory, 256GB SSD, Jet Black, Mag	Computers&Tablets	0.333338889	4.6	110.9919482
72	2-in-1 15.6" Touch-Screen Laptop, Intel Core i7, 12GB RAM, 512GB S	Computers&Tablets	0.1818198347	4.7	117.0657893
73	20.7" LED FHD Monitor	Computers&Tablets	0.3994994995	4.6	113.1257678
74	24" LED FHD Monitor, Black	Computers&Tablets	0.3200128005	4.8	135.0340872
75	27" IPS LED FHD FreeSync Monitor, 27f	Computers&Tablets	0.5600224009	4.8	144.2725127
76	27" LED QHD G-Sync Monitor, Black	Computers&Tablets	0.4166736112	4.6	113.6722674
77	28" LED 4K UHD Monitor, UE590 Series	Computers&Tablets	0.2333411114	4.5	97.43050729
78	31.5" IPS LED FHD Monitor	Computers&Tablets	0.5000166672	4.6	116.2825211
79	32" LED QHD Monitor	Computers&Tablets	0.4473801942	4.7	126.6989659
80	Color Wireless All-in-One Printer	Computers&Tablets	0.3473274435	4.6	111.4469398
81	Wireless All-in-One Instant Ink Ready Printer	Computers&Tablets	0.5555864215	4.5	106.7578186
82	Wireless All-in-One Printer	Computers&Tablets	0.4615739672	4.1	69.8038078
83	Wireless All-in-One Printer	Computers&Tablets	0.4000160006	4.6	113.1422473
84	Wireless All-in-One Printer	Computers&Tablets	0.7143877697	4.1	74.41998573
85	Wireless Color All-in-One Printer	Computers&Tablets	0.5143004086	4.2	78.24182331
86	10.1" Tablet, 32GB	Computers&Tablets	0.400040004	3.7	46.00051632
87	12.3" Tablet, 64GB	Computers&Tablets	0.4387429843	4.1	69.369009



1	Product (Item)	Department	Percentage discount in price	Customer Rating (1 - 5)	Desirability
88	40" 1080p Smart LED HDTV, 5 Series	TV&Home Theater	0.3077041425	4.7	121.7375751
89	43" 4K UHD HDR Smart LED TV, 6 Series	TV&Home Theater	0.1785778064	4.7	116.9428104
90	32" 720p LED HDTV	TV&Home Theater	0.4117889288	4.7	125.455838
91	32" 720p Smart LED HDTV Roku TV, 3 Series	TV&Home Theater	0.07692899454	4.7	113.0155267
92	32" LED 720p Smart TV, H5500 Series	TV&Home Theater	0.4666977799	4.5	104.2799755
93	50" 4K UHD HDR Smart LED Roku TV	TV&Home Theater	0.1071466838	4.6	103.3303832
94	50" 4K UHD HDR Smart LED TV, 7 Series	TV&Home Theater	0.1250031251	4.7	114.8904036
95	50" 4K UHD HDR Smart LED TV, NU6900 Series	TV&Home Theater	0.151519743	4.6	104.8806323
96	55" 4K UHD HDR Smart LED Roku TV, 4 Series	TV&Home Theater	0.1250039064	4.6	103.9571635
97	55" 4K UHD HDR Smart LED TV, NU6900 Series	TV&Home Theater	0.1315824101	4.6	104.1870743
98	55" 4K UHD HDR Smart LED TV, UK6090PUA Series	TV&Home Theater	0.2500062502	4.7	119.6209164
99	55" 4K UHD HDR Smart LED TV, X800G Series	TV&Home Theater	0.2857183674	4.7	120.9358205
100	55" 4K UHD HDR Smart OLED TV, A8G Series	TV&Home Theater	0.1666675926	4.8	128.7411827
101	65" 4K UHD HDR Smart LED Roku TV, 4 Series	TV&Home Theater	0.1111135803	4.5	93.62349234
102	65" 4K UHD HDR Smart LED TV, 7 Series	TV&Home Theater	0.1428591837	4.7	115.5787217
103	65" 4K UHD HDR Smart LED TV, H6500F Series	TV&Home Theater	0.4000080002	4.1	68.62408707
104	65" 4K UHD HDR Smart LED TV, NU6900 Series	TV&Home Theater	0.1041688369	4.6	103.2254768
105	65" 4K UHD HDR Smart LED TV, X800G Series	TV&Home Theater	0.2222246914	4.7	118.5870249
106	65" 4K UHD HDR Smart LED TV, X900F Series	TV&Home Theater	0.1538473373	4.8	128.2000064
107	65" 4K UHD HDR Smart OLED TV, A8G Series	TV&Home Theater	0.2000008	4.8	130.1370863
108	65" 4K UHD HDR Smart QLED TV, Q60 Series	TV&Home Theater	0.09090991736	4.7	113.5640834
109	65" 4K UHD HDR Smart QLED TV, Q70 Series	TV&Home Theater	0.1428581633	4.7	115.5786824
110	65" 4K UHD HDR Smart QLED TV, Q80 Series	TV&Home Theater	0.0555558642	4.7	112.1716028
111	70" 4K UHD HDR Smart LED TV, 6 Series	TV&Home Theater	0.08333472225	4.1	62.16235853
112	75" 4K UHD HDR LED Smart TV, X800G Series	TV&Home Theater	0.2142872449	4.8	130.730489
113	75" 4K UHD HDR Smart LED TV, NU6900 Series	TV&Home Theater	0.1666685185	4.6	105.4043974
114	75" 4K UHD HDR Smart QLED TV, Q60 Series	TV&Home Theater	0.07692366864	4.8	124.9012419
115	75" 4K UHD HDR Smart QLED TV, Q70 Series	TV&Home Theater	0.09090950413	4.7	113.5640672
116	85" 4K UHD HDR Smart LED TV, X900F Series	TV&Home Theater	0.1333337778	4.8	127.3290305
117	4K Ultra HD Blu-Ray Player	TV&Home Theater	0.5000250013	4.6	116.2827788
118	Streaming 4K Ultra HD Audio Wi-Fi Built-In Blu-Ray Player	TV&Home Theater	0.5000166672	4.7	128.5120606
119	Streaming 4K Ultra HD Hi-Res Audio Wi-Fi Built-In Blu-Ray Player	TV&Home Theater	0.3333444448	4.6	110.9921293
120	Streaming 4K Ultra HD Hi-Res Audio Wi-Fi Built-In Blu-Ray Player	TV&Home Theater	0.4800192008	4.8	141.2704101
121	Streaming Audio Blu-Ray Player	TV&Home Theater	0.2857551079	4.5	99.01497635
122	Streaming Audio Wi-Fi Built-In Blu-Ray Player	TV&Home Theater	0.3750468809	4.1	68.13913481
123	1TB Fortnite Neo Versa Console Bundle - Jet Black	Video Gaming	0.233111037	4.8	131.5079565
124	1TB NBA 2K20 Bundle - Black	Video Gaming	0.4000080002	4.8	138.1919411
125	1TB Star Wars Jedi: Fallen Order Deluxe Edition Console Bundle	Video Gaming	0.4000080002	4.8	138.1919411
126	32GB Console - Gray Joy-Con + 2 more items	Video Gaming	0.09411986164	4.6	102.8706436
127	Gamer Master Gaming Desktop, AMD Ryzen 3 2300X, 8GB Memory	Video Gaming	0.2000033334	4.7	117.7530006
128	Gamer Master Gaming Desktop, AMD Ryzen 5 3600, 8GB Memory	Video Gaming	0.2682959548	5	162.3830193
129	Gamer Supreme Liquid Cool Gaming Desktop, AMD Ryzen 7 3700X	Video Gaming	0.300002	5	163.9502954
130	Gaming Desktop, Intel Core i5-9400F, 8GB RAM, NVIDIA GeForce G	Video Gaming	0.2352968858	4.7	119.0747183
131	Gaming Desktop, Intel Core i7-9700K, 16GB RAM, NVIDIA GeForce	Video Gaming	0.1176477509	4.7	114.6056097
132	15.6" Gaming Laptop, AMD Ryzen 5, 8GB Ram, NVIDIA GeForce GTX	Video Gaming	0.4375054688	5	170.5620428
133	15.6" Gaming Laptop, Intel Core i5, 8GB RAM, NVIDIA GeForce GTX	Video Gaming	0.3409129649	4.3	82.4075527
134	15.6" Gaming Laptop, Intel Core i7, 32GB RAM, NVIDIA GeForce RTX	Video Gaming	0.2222234568	4.6	107.3019357
135	17.3" Gaming Laptop, Intel Core i7, 16GB RAM, NVIDIA GeForce GTX	Video Gaming	0.3571454082	4.4	91.50547588

7.2 Acknowledgements and References

- All graphs were generated using Tableau Software and Microsoft Excel.
- All diagrams are self-drawn or are obtained from the IMMC 2020 Problem Statement.
- Kuklin, P. (n.d.). Crowd Control Management Strategies: 5 Techniques for Safer, Happier Customers.Retrieved March 18, 2020, from https://www.lavi.com/en/resources-detail/crowd-control-techniques
- McCarthy, D. (2020, March 10). Top Crowd Control Tips & Best Practices for Your Venue. Retrieved March 18, 2020, from https://www.socialtables.com/blog/event-venues/crowd-control-tips/

7.3 Code

Below is our code written in C++ used to generate the allocation of goods, using the Genetic Algorithm as explained in section 4.3.3.

```
#include <bits/stdc++.h>
using namespace std;

struct product {
    int major_cat=-1, index, quantity=0;
    long double price=0, di=0, desirability=0;
    //product(int a, int b, long long double c, long long double d, long long double
e, long long double f): major_cat(a), index(b), price(c), F(d), M(e), desirability(f)
{}
    long double calculate_d(){ // Probability
        //long long double dummy = log(F/M + exp(1.00));
        return 1.00 - (1.00/di);
    }
};

long double calculate_lambda(vector <product> A, long double width, long double
max_d){ // Product, quantity on shelf
    long double sum = 0;
    for (int i = 0; i < A.size(); ++i){
        product B = A[i];
        int quantity = A[i].quantity;
        sum += B.calculate_d() * B.price * (long double)quantity;
        max_d = max(max_d, B.desirability);
    }
     return (long double)( (sum * max_d * 1.7) / (width*width) ); // Note: width is
the true length of the width
}


product xx,null_product;

string ss,prev_dept;

int max_capacities[7]={224,12,30,6,245,272,78};

pair<int,vector<vector<product> > > population[20];
vector<int> v1;
vector<product> v2,to_push;
//for each population there's 7 departments

bool cmp(pair<int,vector<vector<product> > >a,pair<int,vector<vector<product> > >b)
{return(a.first>b.first);}

inline void get_cell(){//reads stuff into string ss...
  char c;
  ss="";
  while(c=getchar()){
    if(c=='\n' || c==','){
      if(ss.length()>0)break;
      else continue;
    }
    ss+=c;
  }
}

long double convert(string s){//convert string to long double...
```

```
  long double ans=0,coeff=0.1;
  int it;
  for(it=0;it<s.length();it++){
    if(s[it]=='.'){
      it++;
      break;
    }
    ans*=10.0;
    ans+=s[it]-'0';
  }
  for(;it<s.length();it++){
    ans+=coeff*(s[it]-'0');
    coeff/=10.0;
  }
  return ans;
}

vector<pair<int,int> > root_list[65];//list of nodes that can go to node i
void read_graph(){
  ifstream input("edge_list.txt");
  int n,k,w,prev,node;
  //vertical aisles...
  input>>n;
  for(int x=0;x<n;x++){
    input>>k>>w;//width of the aisle...
    for(int y=0;y<k;y++){
      input>>node;
      if(y>0){
        root_list[node].emplace_back(prev,w);
      }
      prev=node;
    }
  }

  //horizontal aisles...
  input>>n;
  for(int x=0;x<n;x++){
    input>>k>>w;//width of the aisle...
    for(int y=0;y<k;y++){
      input>>node;
      if(y>0){
        root_list[node].emplace_back(prev,w);
      }
      prev=node;
    }
  }
  input.close();
}

//genetic algo...................................................
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());
inline int rand(int x, int y) {return (rng() % (y+1-x)) + x;}

long double ans=0;
bool vis[65];
vector<product> items[65],empty;//items in each node...

void backtrack(int node,long double max_d=0){
  vis[node]=true;
  for(int x=0;x<items[node].size();x++)max_d=max(max_d,items[node][x].desirability);
  for(int x=0;x<root_list[node].size();x++){
```

```
                ans+=calculate_lambda(items[root_list[node][x].first],root_list[node]
[x].second,max_d);
  }
  for(int x=0;x<root_list[node].size();x++){
    if(vis[root_list[node][x].first])continue;
    backtrack(root_list[node][x].first,max_d);
  }
}

long double find_fitness(vector<vector<product> > &perm){//each way to permute the
goods...
  for(int x=0;x<65;x++){
    items[x].clear();
    vis[x]=0;
  }
  int arr1[9]={1,2,3,10,11,12,19,20,21},tmp;
  for(int x=0;x<perm[0].size();x++){//appliances...
    if(perm[0][x].index==-1)continue;
    if(x<44)items[26].push_back(perm[0][x]);
    else if(x>=(44+(9*16))){
      items[30].push_back(perm[0][x]);
    }else{
      items[ arr1[(x-44)/16] ].push_back(perm[0][x]);
    }
  }
  for(int x=0;x<perm[1].size();x++){//audio
    if(perm[1][x].index==-1)continue;
    if(x<6){//left shelve...
      items[5].push_back(perm[1][x]);
    }else items[9].push_back(perm[1][x]);
  }
  for(int x=0;x<perm[2].size();x++){//cameras
    if(perm[2][x].index==-1)continue;
    if(x<15)items[54].push_back(perm[2][x]);
    else items[55].push_back(perm[2][x]);
  }
  for(int x=0;x<perm[3].size();x++){//cell phones
    if(perm[3][x].index==-1)continue;
    if(x<3)items[2].push_back(perm[3][x]);
    else items[3].push_back(perm[3][x]);
  }
  for(int x=0;x<perm[4].size();x++){//computers & tablets...
    if(perm[4][x].index==-1)continue;
    if(x<54)tmp=24;
    else if(x<54+15)tmp=9;
    else if(x<54+15+48)tmp=25;
    else if(x<54+15+48+36)tmp=51;
    else if(x<54+15+48+36+36)tmp=60;
    else tmp=58;
    items[tmp].push_back(perm[4][x]);
  }
  for(int x=0;x<perm[5].size();x++){//tv...
    if(perm[5][x].index==-1)continue;
    if(x<140){
      items[min(28,26+(x/28) )].push_back(perm[5][x]);
    }else{
      tmp=29+((x-140)/36);
      if(((x-140)%36)<6);
      else if(((x-140)%36)<6)tmp++;
      else tmp+=2;
      items[tmp].push_back(perm[5][x]);
    }
  }
  for(int x=0;x<perm[6].size();x++){//video games...
    if(perm[6][x].index==-1)continue;
    if(x>=60)items[14].push_back(perm[6][x]);
    items[(x/12)+5].push_back(perm[6][x]);//reach the node on top...
  }

  ans=0;
  backtrack(1);
  return ans;
}

int main(){
  freopen("data_sheet.csv","r",stdin);

  vector<string> row;
  vector<vector<string> > table;
  int it=0,sum=0,cur;

  null_product.index=-1;//no product on the shelve...

  //read the csv file...
  for(int i=0;i<134;i++){
    for(int x=0;x<14;x++){
      get_cell();
      row.push_back(ss);
    }get_cell();//read past the remaining stuff...

    if((i==0 || row[0]!=prev_dept)&&i!=1){
      population[0].second.push_back(to_push);
      //cout<<sum<<'\n';
      sum=0;
    }
    prev_dept=row[0];

    xx.major_cat=population[0].second.size();
    xx.index=i;
    xx.price=convert(row[4]);
    xx.desirability=convert(row[8]);
    xx.di=convert(row[13])+1.0;
    xx.quantity=int(0.64/convert(row[11]));//number of items in 1 unit square...

    table.push_back(row);
    row.clear();

    cur=convert(row[12]);//no. of unit squares occupied...
    for(int x=0;x<cur;x++)population[0].second.back().push_back(xx);//item i...
    //create copies of the product...
    sum+=cur;
  }

  for(int i=0;i<7;i++){
    while(population[0].second[i].size()<max_capacities[i]){
      population[0].second[i].push_back(null_product);
    }
    //cout<<population[0].second[i].size()<<'\n';
  }

  //read the list of edges...
  read_graph();
  for(int i=1;i<20;i++)population[i]=population[i-1];
  //shuffle allocations...
  for(int i=0;i<20;i++){
    for(int x=0;x<7;x++){
      shuffle(population[i].second[x].begin(),population[i].second[x].end(),rng);
    }
  }

  for(int g=0;g<10000;g++){//generations...
    for(int i=0;i<20;i++){
      population[i].first=find_fitness(population[i].second);
    }
    sort(population,population+20,cmp);//sort from worst to best
    cout<<population[19].first<<'\n';
    for(int i=0;i<10;i++){
      population[i]=population[i+10];//delete the worse 10
      for(int x=0;x<population[i].second.size();x++){//each department...
        v1.clear();v2.clear();
        for(int y=0;y<population[i].second[x].size();y++){
          if(rand(1,35)==1){//shuffle 1/10 of the shelves within each department...
            v1.push_back(y);//remember the indexes...
            v2.push_back(population[i].second[x][y]);//the item...
          }
        }
        if(v2.size()==0)continue;
        shuffle(v2.begin(),v2.end(),rng);

        for(int y=0;y<v1.size();y++){
          population[i].second[x][ v1[y] ]=v2[y];
        }
      }
    }
  }

  for(int i=0;i<20;i++){
    population[i].first=find_fitness(population[i].second);
  }
  sort(population,population+20,cmp);//sort from worst to best

  ofstream output("output.txt");//output the allocation of goods...

  for(int i=0;i<7;i++){
    for(int x=0;x<population[19].second[i].size();x++){
      output<<population[19].second[i][x].index<<' ';
    }
    output<<"\n\n";
  }

  output.close();
  return 0;
}
```

7.4 Full allocation of goods

Below is the list of numbers that represent the item ID (based on their index in the original data sheet, where the ID of the first good is 0) of each good to put in each *unit square*, with the original floor plan. The sections are in the order of departments: Appliances, Audio, Cameras, Cell Phones, Computers & Tablets, TV & Home Theater, Video Games.

The order of the numbers express which shelve to place each item on, within the department. The value -1 represents an empty space. The following allocation of goods with the original floor plan was computed with the Genetic Algorithm, with a mutation rate of $\frac{1}{35}$:

Appliances: 28 12 11 12 0 13 13 2 30 23 1 18 17 9 25 17 17 28 3 25 30 18 11 25 6 7 15 4 -1 26 29 12 6 12 16 15 30 1 28 16 21 25 17 0 12 5 22 25 13 24 1 9 24 22 12 12 15 25 25 27 20 26 9 13 27 11 26 28 3 14 14 15 24 22 22 24 29 5 16 27 23 8 29 3 23 16 15 27 4 22 29 29 11 6 17 13 6 16 2 22 10 25 0 0 28 9 11 -1 2 4 15 2 12 21 22 21 10 21 8 10 8 -1 2 5 27 20 19 24 3 3 11 16 8 12 6 10 9 1 -1 -1 28 14 10 12 10 30 6 4 26 7 6 11 2 5 5 24 7 16 15 8 2 7 0 21 12 23 10 22 18 16 4 18 15 12 1 7 19 2 20 11 -1 23 12 12 3 19 14 21 14 6 10 24 9 28 11 -1 21 5 23 9 30 19 21 9 24 12 5 10 26 21 3 19 5 23 21 28 10 20 -1 18 11 23 3 20

Audio: 31 31 32 31 32 32 -1 -1 -1 -1 -1 -1

Cameras: 42 42 37 40 40 -1 42 37 -1 38 38 38 -1 -1 37 41 36 41 34 33 41 33 36 33 39 39 36 35 35 34

Cell Phones: 45 44 -1 -1 -1 43

Computers&Tablets: 72 74 77 75 75 72 75 72 72 77 75 75 71 75 71 74 75 72 73 73 74 75 72 73 76 76 73 73 76 77 72 77 72 -1 74 77 77 72 71 73 75 75 72 75 79 75 72 76 76 77 77 82 77 79 51 84 63 50 78 50 54 57 57 56 51 56 53 78 48 74 80 79 47 73 81 76 79 72 71 79 72 73 74 82 48 79 83 76 75 73 82 46 76 77 72 82 49 74 83 76 82 81 82 49 71 74 80 73 73 46 73 47 83 72 46 77 71 74 47 76 78 81 49 71 76 47 50 80 71 80 46 50 83 46 75 81 47 74 83 46 74 77 81 47 46 47 49 76 80 75 80 74 83 61 53 67 52 81 48 66 62 66 55 62 61 52 62 70 66 62 48 51 58 68 69 64 58 68 63 56 78 65 61 50 54 69 58 52 60 60 62 48 65 78 53 65 55 54 56 46 54 59 59 78 58 59 52 48 64 60 57 55 61 49 57 85 85 58 59 84 61 70 53 67 69 62 60 47 84 63 52 54 48 55 68 64 78 57 51 48 85 70 78 50 60

TV & Home Theater: -1 90 94 100 102 103 89 99 89 93 91 96 102 -1 91 91 95 100 97 106 113 103 113 88 113 94 94 102 108 104 110 112 108 101 90 91 107 102 105 112 103 105 99 102 113 -1 100 104 106 107 99 105 108 107 105 106 93 100 100 109 104 110 110 102 106 114 88 99 109 110 99 114 99 107 103 104 108 105 104 101 112 108 102 108 91 108 108 -1 109 114 101 105 106 90 101 113 105 111 106 -1 108 103 101 100 111 106 109 105 104 103 102 103 105 112 106 104 108 106 113 112 112 100 104 100 107 101 99 93 100 109 111 106 112 104 113 109 107 100 103 99 86 98 115 117 95 116 109 90 101 100 106 99 107 111 106 102 111 103 114 112 103 104 102 105 110 112 114 114 92 102 112 113 106 114 105 103 94 101 112 112 109 109 119 98 92 120 94 98 117 92 98 86 97 95 87 88 118 97 98 106 105 89 120 107 87 98 87 103 90 95 97 95 110 109 102 110 110 99 90 95 87 93 114 112 108 108 94 109 86 96 88 93 93 108 89 113 92 108 91 104 113 100 107 92 -1 86 104 108 87 119 92 96 96 106 109 -1 111 113 109 112 109 109 112 89 113 107 99 99 111 107 111 107

Video Games: 129 132 124 129 129 133 128 124 133 132 133 124 128 128 132 123 126 132 122 122 126 133 129 133 126 127 121 127 130 127 132 128 130 132 121 122 133 126 130 132 128 123 122 129 132 126 126 129 121 121 123 129 124 130 123 127 122 129 130 127 128 128 125 125 131 131 128 125 131 131 131 125 125 125 125 -1 131 125