MGSC670 Revenue Management
The Retailer Markdown Game

Group Members: Kexin Wang (260787422), Yichen Wang (260761601), Yingxin Jiang (261007353), Yulin Hong(260898713)

1. Introduction

In the retailer game, each round starts with 2000 items and the sales season is 15 weeks. One needs to choose their strategy during the 15 weeks sale time. One has 4 possible strategy: $60, $54, $48 and $36. Each time after the price decreases, one cannot choose to increase the price. For example, if at week 6 one changes the sale price from $60 to $54, they cannot change back to $60 anymore. The only options left are to maintain the price at $54, or decrease the price further to $48 or $36.

In this assignment, we aimed to develop a strategy to achieve a good performance in the retail markdown game under different random circumstances.

2. Observation and Preparation

We visualized this problem as an optimization problem. In this game, what we want is to maximize the revenue after 15 weeks of sale and we need to determine how many weeks we should put an item on sale for a certain price.

To model this problem using optimization solver, we first utilized the original dataset to generalize the increment in demand after decreasing the price. Below is our result:

Item	avg sale for $60	avg sale for $54	avg sale for $48	avg sale for $36	sale increase rate $60 to $54	sale increase rate $60 to $48	sale increase rate $60 to $36
1	58.333	76.000	na	na	1.303		
2	107.667	144.000	na	na	1.337		
3	59.333	82.333	na	na	1.388		
4	61.167	77.889	na	na	1.273		
5	92.500	113.667	na	na	1.229		
6	114.143	na	150.125	na		1.315	
7	67.429	na	119.625	na		1.774	
8	53.000	na	96.750	na		1.825	
9	73.714	na	131.875	na		1.789	
10	67.286	na	97.125	na		1.443	
11	100.444	na	na	182.667			1.819
12	64.111	na	na	188.500			2.940
13	65.556	na	na	196.667			3.000
14	61.444	na	na	164.000			2.669
15	62.333	na	na	175.000			2.807
				avg increse rate	1.306	1.629	2.647

Based on our calculation, we found that the average increment in sales when one decrease the price from $60 to $54 is 1.306, from $60 to $48 is 1.629 and from $60 to $36 is 2.647. Since demand increment is homogeneous across different items, we will use these numbers to forecast the increment in sales after we decrease the price.

3. Methodology

To find the optimal time length for each price promotion, we used Gurobi to build the model and perform the optimization. The variables we are going to use is shown below:

```
d_60 = model.addVar(lb=0,ub=106,vtype=GRB.INTEGER,name="d_60")
x_60 = model.addVar(lb=0,ub=15,vtype=GRB.INTEGER,name="x_60")
x_54 = model.addVar(lb=0,ub=15,vtype=GRB.INTEGER,name="x_54")
x_48 = model.addVar(lb=0,ub=15,vtype=GRB.INTEGER,name="x_48")
x_36 = model.addVar(lb=0,ub=15,vtype=GRB.INTEGER,name="x_36")
```

D_60 refers to the weekly demand at $60, where the weekly demand for price $54, $48 and $36 can be represented in terms of d_60 since we have obtained the average incremental coefficient for each price decrease. For example, the weekly demand at price p = $54 can be expressed as d_54 = 1.306*d_60 as the price reduction will lead to a 30.6% increase in weekly demand. Its lower bound is 0 and upper bound is defined in two methods, leading to two different model versions. The first upper bound is defined by the average of the max $60 sales of the 15 items given in the dataset ($106), whereas the second version defines the upper bound the max value among the average values of $60 sales for each 15 items ($114).

X_60, x_54, x_48 and x_36 refers to the number of weeks the item is sold at a specific price (x_60 refers to the number of weeks the item is sold at $60 and the rest are defined similarly). The lower bound is set to be 0 and the upper bound is set to 60 as it is the length of the trial.

Our objective function is defined as below:

```
exp = 60*d_60*x_60+54*1.306*d_60*x_54+48*1.629*d_60*x_48+36*2.647*d_60*x_36
model.setObjective(exp, GRB.MAXIMIZE)
```

The expected value is the total revenue expected, and the objective function aims to maximize its value. Several constraints are added according to the circumstances. First is that the total sales should be less than 2000, which is equivalent to the inventory as sales

could not surpass inventory. Second is that the sum of weeks corresponds to different prices should not surpass 15 weeks. Finally, the number of weeks corresponding to $60 should be at least one, meaning week 1 will have full price no matter what.

4. Final Strategy

From the results generated from Gurobi and our manual tests that were recorded in EXCEL, we find that the combination of 2 weeks with $60, 13 weeks with $54 would have the lowest difference between revenue with the perfect foresight strategy and revenue with Gurobi result. The results were still better as compared to the other combinations, therefore, the optimal strategy to employ would be using Gurobi to implement LInear Optimization (2 week $60, 13 weeks $54).

5. Conclusion

In the retail industry the implementation of markdowns to optimize profit and unit sales is imperative for the practice of revenue management in this field. These reports describe the effectiveness of Gurobi algorithms on The Retail Markdown Game and it is the best one to employ that has the lowest difference between revenue with perfect foresight strategy and revenue with our Gurobi result.