MIDDLE EAST TECHNICAL UNIVERSITY

Term Project Report

IE407

January 13, 2021

Students:
John Doe

Contents

1 Introduction **2**

2 Project Body **2**

2.1 Proposed Model . . . 3

2.1.1 Q2 . . . 5

2.1.2 Q3 . . . 6

2.1.3 Q4 . . . 9

2.1.4 Q5 . . . 9

2.1.5 Q6 . . . 9

2.1.6 Q7 . . . 10

2.2 Discussion . . . 10

3 Conclusion **11**

4 Appendix **11**

4.1 Data Set for Q1,Q2 . . . 11

4.2 Rules . . . 13

1 Introduction

In this project, we have developed a solution for "Assortment Selection and Shelf Space Allocation Problem". Although in the literature, the problem has different variations, we studied a simplified version of the problem where,

- If a product is chosen for the assortment, then all the facings of the product have to be filled fully. This restriction is especially important since dealing with variable size allocations makes the problem more complex. Such situation can be found in [1]
- The manager does not want to allocate more than four facings for a product. The importance of this constraint is explained in 2
- Other constraints can be found in 4.2

2 Project Body

Initially, we approached the problem by defining two decision variables:

X_{ik} : Will the product i be on shelf k. Such that,

$$X_{ik} = \begin{cases} 1, & if\ i\ is\ allocated\ in\ k \\ 0, & otherwise \end{cases}$$

f_{ik}: Number of allocated facings for product i on shelf k.

With these decision variables, the objective function (since our aim is maximizing the profit) became:

$$max \sum_{i=1}^{|I|} \sum_{k=1}^{|K|} \pi_i \gamma_k d_i (f_{ik} * b_i)^{\beta_i}$$

As can be confirmed, if we choose f_{ik} as a decision variable then the given objective function makes the model non-linear. Since the "opensolver" is not good at solving non-linear problems, we decided to modify the initial model to obtain a linear model.

The proposed model 2.1 has been built considering (with the help of) Rule

2. Since for a product we can not allocate more than 4 facings (which gives us a constant domain for facings), defining a seperate decision variable for facings is unnecessary. Rather we define a new decision variable instead of the previous ones,

X_{ijk} : Will the product i have j allocations on shelf k. Such that,

$$X_{ik} = \begin{cases} 1, & if\ i\ is\ allocated\ j\ times\ in\ k \\ 0, & otherwise \end{cases}$$

where $j \in \{1, 2, 3, 4\}$. With this new approach, we have achieved to develop a linear model which can be reached in 2.1.

2.1 Proposed Model

Parameters:
I : Set of products
I^1: Set of product pairs (i1, i2) that cannot be placed on the same shelf.
I^2: Set of product pairs (i1, i2) that will be included together in the assortment.
K : Set of shelves
π_i: Profit made by selling one unit of product i
w_k : Width of shelf k
γ_k : Shelf k's effect on demand
ds_k : Depth of shelf k
dp_i : Depth of unit product i
b_i: Width of a facing for product i
d_i: Coefficient for demand rate for product i per unit width and one facing
β_i: Space elasticity factor for product i
s_i^l : Lower bound on the shelf inventory of product, if i is selected in the assortment
s_i^u : Upper bound on the shelf inventory of product, if i is selected in the assortment

Notation:
$|I|$: Size of the set of products
$|K|$: Size of the set of shelves

Calculated Data Sets:
N_{ik}: number of product i in shelf k per allocation:
$N_{ik} = \lfloor ds_k/dp_i \rfloor$

Decision Variables:
X_{ijk} : Will the product i have j allocations on shelf k. Such that,

$$X_{ik} = \begin{cases} 1, & if\ i\ is\ allocated\ j\ times\ in\ k \\ 0, & otherwise \end{cases}$$

Objective function:

$$max \sum_{i=1}^{|I|} \sum_{k=1}^{|K|} \sum_{j=1}^{4} \gamma_k * \pi_i * d_i * (b_i)^{\beta_i} * (j)^{\beta_i} * x_{ijk}$$

Subjected To:

$$\sum_{k=1}^{|K|} \sum_{j=1}^{4} X_{ijk} \leq 1 \quad \forall i \in I \qquad (Rule\ 1)$$

$$\sum_{k=1}^{|K|} (N_{ik} * (\sum_{j=1}^{4} X_{ijk} * (j)) - s_i^u * (\sum_{j=1}^{4} X_{ijk})) \leq 0 \quad \forall i \in I \tag{1}$$

$$\sum_{k=1}^{|K|} (N_{ik} * (\sum_{j=1}^{4} X_{ijk} * (j)) - s_i^l * (\sum_{j=1}^{4} X_{ijk})) \geq 0 \quad \forall i \in I \tag{2}$$

Constraint (1) and (2) come from Rule 3 and 4.

$$-w_k + \sum_{i=1}^{|I|} (b_i * (\sum_{j=1}^{4} X_{ijk} * (j))) \leq 0 \quad \forall k \in K \qquad (Rule\ 6)$$

$$(\sum_{k=1}^{|K|} \sum_{j=1}^{4} X_{i_1 jk}) - (\sum_{k=1}^{|K|} \sum_{j=1}^{4} X_{i_2 jk}) = 0 \quad \forall (i_1, i_2) \in I^2 \qquad (Rule\ 7)$$

$$(\sum_{j=1}^{4} X_{i_1 jk}) + (\sum_{j=1}^{4} X_{i_2 jk}) \leq 1 \qquad \forall k \in K \quad \forall (i_1, i_2) \in I^1 \qquad (Rule\ 8)$$

Restrictions:

$$X_{ijk} \in \{0,1\} \qquad \forall i \in I \quad \forall j \in \{1,2,3,4\} \quad \forall k \in K$$

2.1.1 Q2

In this section, solution of the problem for given data set 4.1 obtained by using the given model and our observations about the results can be found. We used below formulas to simplify our calculations:

$$1 - f(i,k)^{\beta_i} = \sum_{j=1}^{4} X_{ijk} * j^{\beta_i} \quad \forall i \in I \quad \forall k \in K$$

$$2 - f(i,k) = \sum_{j=1}^{4} X_{ijk} * j \quad \forall i \in I \quad \forall k \in K$$

$$3 - Z(i,k) = \sum_{j=1}^{4} X_{ijk} \quad \forall i \in I \quad \forall k \in K$$

Important Notes: 1st formula is used in objective function.
2nd and 3rd formulas are used repeteadly on constraints.
1st formula was selected like that to make our model lineer.

Objective function result was found as: **6838,349047**

According to results from opensolver:
178,9355445 from products in shelf 1 which has γ_k of 0.25 and width of 50cm
1105,538071 from products in shelf 2 which has γ_k of 0.6 and width of 65cm
3508,467561 from products in shelf 3 which has γ_k of 1 and width of 80cm
1800,355036 from products in shelf 4 which has γ_k of 0.6 and width of 95cm
345,0528341 from products in shelf 5 which has γ_k of 0.25 and width of 110cm

3rd shelf that has the highest γ_k value gained most money.
Also we gained more from shelf 4 in comparison to shelf 2 and
5 in comparison to shelf 1 even though they had the same γ_k.
The reason of this is the difference in their width.
Shelves 4 and 5 had also more space for more products and facings.

Because of that those shelves gained more money even though they had the same γ_k.

2.1.2 Q3

According to opnesolver objective function result was found as: **37906,15197**

In our model we are using **4*ProductCount*ShelfCount** decision variables(500 for smaller, 2000 for bigger data set).
While the lesser data set worked in an instant larger data set took 34 seconds to finish. This is because number of decision variables increases, which in turn increases the time it takes to solve relaxation of a problem and number of problems(increase in number of problems is the main reason).

For a problem like this the worst case is complete enumeration of decision variables. And all of our decision variables are binary which means normally for the worst case we would have needed to solve 2^{500} for lesser data set and 2^{2000} for bigger data set. but because of our first constraint for any i and k values we would have 5 cases which are:

	X_{i1k}	X_{i2k}	X_{i3k}	X_{i4k}
case1:	0	0	0	0
case2:	1	0	0	0
case3:	0	1	0	0
case4:	0	0	1	0
case5:	0	0	0	1

That means we have only 5 different cases for every 4 decision variables instead of 2^4 Because of that in the case of complete enumeration we would have 5^{100} problems for smaller data set and 5^{400} for bigger data set (these numbers were only calculated using 1st group of constraints. By adding other constrains we can reduce it more). As we can see for the worst case(complete enumeration) number of problems that we need to solve increases exponentially(There would have been 5^{300} times more problems which we would need to solve if we only had 1st group of constraints). Because of that total solution time will also increase exponentially.

Since we are using more constraints and decision variables, the size of the matrices we are gonna need the LP relaxation of a problem, will also increase in size.

Lets calculate sizes of these matrixes for each data set:

1- Smaller data set:
400 decision variables
25 constrains from 1st group of constraints. Which also adds 25 slack variables(<=)
25 constrains from 2nd group of constraints. Which also adds 25 excess variables(>=)
25 constrains from 3rd group of constraints. Which also adds 25 slack variables(<=)
5 constrains from 4th group of constraints. Which also adds 5 slack variables(<=)
4 constrains from 5th group of constraints. (=)
15 constrains from 6th group of constraints. Which also adds 15 slack variables(<=)
so we have 99 constrains
and 495 decision variables
495-99=396

That means size of our Matrices are:
B=99*99
N=99*396
x_B=99*1
x_N=396*1
c_B=99*1
c_N=396*1
b=99*1
B^{-1}=99*99

2- Larger data set:
2000 decision variables
100 constrains from rule 1st group of constraints. Which also adds 100 slack variables(<=)
100 constrains from rule 2nd group of constraints. Which also adds 100 ex-

cess variables(>=)
100 constrains from rule 3rd group of constraints. Which also adds 100 slack variables(<=)
5 constrains from rule 4th group of constraints. Which also adds 5 slack variables(<=)
13 constrains from rule 5th group of constraints. (=)
60 constrains from rule 6th group of constraints. Which also adds 60 slack variables(<=)
so we have 378 constrains
and 2365 decision variables
2365-378=1987

That means size of our Matrices are:
B=378*378
N=378*1987
x_B=378*1
x_N=1987*1
c_B=378*1
c_N=1987*1
b=378*1
B^{-1}=378*378

Note: Those numbers are calculated while ignoring artificial variables
.
Since we are going to use simplex method we would need to make matrix multiplications.
Which are the most costly operations in big data sets.
(if you are making opearation: AxB for A matrix being a*b and B matrix b*c you would need to do a*b*c multiplications)

The costliest operation we are going to need to do in Simplex method is:
$c_B^T * B^{-1} * N$ (its actually: $c_N^T - c_B^T * B^{-1} * N$(reduced cost vector) but matrix multiplication is much slower than matrix differention so it doesnt make almost any difference)

For first data set the number of operations this multiplication will take is:
(1*99*99)*(1*99*396)=384238404 operations
For larger data set the number of operations this multiplication will take is:

(1*378*378)*(1*378*1987)=107318172024 operations

283910508/3881196=279.30100403

so as we can easily see, the time it takes to calculate this operation has increased more than 275 times. But since difference between other operations won't be as big as this operation, rate of time it takes to solve these data sets will be a little less then this number. Which is still a great increase but nowhere near the increase on number of problems.

So in the end, solution time of our problem will grow exponentially because of the increase in problem count.

2.1.3 Q4

In this section, we analysed how the solution given in 2.1.1 changes if we add the 6th shelf with $w_6 = 40$, $ds_6 = 25$, $\gamma_6 = 0.45$ to the K given in 4.1.

2.1.4 Q5

In this section, we analysed how the solution and profit given in 2.1.1 change, if the shortest (in width) available shelf is allocated to some other category. The shortest shelf in 4.1 is the one with Shelf number 1. Since the shelves are allocated for only the products belong to the same category, by allocating it to some other category, in fact we remove it from our data set.

2.1.5 Q6

In this section, we analysed how the profit found in 2.1.1 changes and what are the basic variables when two seperate cases happen:

- the width of Shelf 5 (w_5) is increased by 5 cm
- the width of Shelf 5 (w_5) is increased by 10 cm

Firstly, for both cases, we are relaxing constraints. Therefore, we can say that our profit will be a little higher, or will be the same.

For the first case, we see that the new profit is x. As a result of increasing shelf width, we gain a y profit increase.

For the second case, we are relaxing constraints a bit more. As a result of increasing shelf width even more, our new profit is z, even t more than the first case, as we expected.

What we can say about our basic variables is ...

Lastly, in classes, we see that as we relax the constraints more, the optimized value will be worse, or the same. And here, we especially see how the constraint changes are affecting the value in practice.

2.1.6 Q7

In this section, we analysed how the solution found in 2.1.1 changes when the restriction requiring Product 3 and Product 8 to be placed on different shelves is removed and the restriction requiring these two products to be assorted together and placed to the same shelf is added.

As a result of this change in constraints, we should see a change in profits probably. Nevertheless, this may not always happen. In the case of the products are never put in the shelves, or the constraint does not really constrain our optimal value, we should see that profit does not change. Otherwise, we still cannot say that this will change profit positively or negatively. The reason is we are removing one constraint, and adding a new constraint. Generally, what we can say is if we are removing a constraint, the profit will be at least as good as before, and vice versa. However, here two things happening, we are removing a constraint, and at the same time adding a new constraint. Before solving the model, it is almost impossible to say whether the profit will increase or decrease.

We removed the constraint in the model and saw that the profit does not change. To inspect the situation, we need to look at the results before the change. From results, we see that in the optimal situation, product 3 is in shelf 3, and product 8 is in shelf 5. Moreover, with the constraint removed, it is still the same. From that, we can conclude that our profits already higher than the case of these two products are on the same shelf. This constraint does not affect our profits.

2.2 Discussion

Discussion...

3 Conclusion

Conclusion

4 Appendix

4.1 Data Set for Q1,Q2

Table 1: Sets of product pairs

I^1 :	(2, 5) , (3, 8), (16, 20)
I^2 :	(1, 12), (3, 8), (9, 15), (16, 20)

Table 2: Product based data

Product number	π_i	b_i	dp_i	d_i	β_i	s_i^l	s_i^U
1	15	10	7	4	0.5	2	15
2	8	9	6	10	0.2	1	19
3	12	5	10	10	0.3	1	23
4	6	7	7	7	0.2	1	9
5	11	9	9	5	0.9	1	16
6	14	6	8	2	0.4	1	21
7	14	9	6	1	0.5	1	11
8	6	5	9	6	0.3	1	18
9	5	9	9	7	0.8	2	11
10	11	10	9	3	0.8	1	11
11	12	7	5	4	0.1	2	17
12	8	5	6	7	0.8	2	22
13	11	7	6	2	0.1	1	12
14	13	8	9	9	0.8	3	12
15	7	9	8	11	0.8	3	19
16	14	23	5	2	0.1	2	16
17	9	25	6	9	0.1	1	10
18	10	17	8	1	0.8	2	16
19	13	15	9	4	0.2	2	20
20	5	23	6	2	0.6	3	19
21	11	19	8	6	0.6	2	24
22	11	19	9	6	0.4	1	16
23	7	16	7	8	0.5	2	13
24	10	14	5	2	0.8	1	16
25	13	16	10	4	0.9	2	14

Table 3: Shelf based data

Shelf number	w_k	ds_k	γk
1	50	34	0.25
2	65	30	0.60
3	80	26	1
4	95	27	0.60
5	110	29	0.25

4.2 Rules

1- **If a product is selected in the assortment, than all facings for the product must be placed on the same shelf.** $\surd$
2- **The manager does not want to allocate more than four facings for a product.** $\surd$
3- **If a product is selected in the assortment, then a minimum shelf inventory amount must be placed on the shelves. Similarly, for each product there is an upper bound on the shelf-inventory.** $\surd$
4- **If a product is selected in the assortment, lower and upper bounds on its facing number are calculated by using shelf depth, product depth and lower and upper bounds on the shelf-inventory.** $\surd$
5- *Each product provides a certain profit per unit sold.*
6- **Each shelf has a certain width and the total width of the facings placed in the shelf cannot exceed its width.** $\surd$
7- **For some pairs of products, there is a restriction that if one is included in the assortment, the other product must also be included.** $\surd$
8- **For some pairs of products, there is a restriction that they cannot be on the same shelf** $\surd$

References

[1] Hübner, A., and Schaal, K. An integrated assortment and shelf-space optimization model with demand substitution and space-elasticity effects. *European Journal of Operational Research 261*, 1 (2017), 302 – 316.