Operation Research Final Report

Elizabeth Anajovich, Celton Do, Ketsaraporn Kaewkhiaolueang, Evan Pinckard, and Bryce Samwel

11/30/2020

Contents

1 Abstract **2**

2 Introduction **3**

3 Literature Review **3**

4 Project Details and Methodology **4**

5 R Model and Results **5**

6 Model Discussion & Future Adaptations **9**

7 Conclusion **10**

8 References **11**

1 Abstract

In the case of a distribution company growing and needing to further expand its warehousing capabilities, they are going to want to do so in an efficient manner to best serve its users when cost is not a primary factor; the allocation of these warehouses is crucial to timely delivery. Linear programming and goal programming were utilized to explore a model adhering to three main goals: raise the capacity floor, lower the capacity ceiling, and maximize total capacity. This varies from many of the reviewed literature pieces as almost all have cost built into them, but they remain relevant to the proposed case in that all are trying to minimize a parameter (cost, travel distance, etc.) while maximizing output in some form (customers served, efficiency of distribution, etc.). The case focused on was broken into meeting the mentioned minimizing and maximizing while working around the constraints of 15 subregions unevenly distributed across 5 regions, along with each having access to 5 different potential building types. The model output multiple optima such that it was altered slightly to more evenly distribute the capacity allocation to subregions 1, 3, and 6. From there, the model was generalized for further application by the individual requesting it such that it could be modified for more personal use.

2 Introduction

As a distribution company grows it starts to face the need to make many different decisions. In the particular case explored by our project, we were provided objectives, conditions, and constraints to model warehouse location optimization for a distributor by Aurobindh Puthanpura. What we will be pursuing and explaining throughout the rest of this paper is how we determined methods, a comparison to other warehousing problems, and our conclusions with respect to our illustrated model output.

By utilizing RStudio we were able to implement LP modeling with goal programming methodology. All work was done in an R Markdown document using the gridExtra, tufte, magrittr, dplyr, ROI, ROI.plugin.glpk, ompr, and ompr.roi packages. The initial data was set-up using Excel spreadsheet to create the explicit model prior to all RStudios R Markdown implementation.

In our case there are 5 regions, with 15 total subregions (not evenly distributed), and 5 different building types that can be constructed. The model was used to determine how many of each building type and where they should be built. The cost of each building was purposefully omitted at the recommendation of Aurobindh Puthanpura as an unnecessary consideration at this time.

3 Literature Review

Warehouse allocation optimization is crucial to any big companies and retailer, especially those who focus on e-commerce to ensure fastest deliveries with cheapest costs. These companies usually have experts that create the optimization, but sometimes they hire an outsider to do it such as AnyLogistix. Their model usually uses data from sold goods, loss from shortages of demanded goods, transportation cost, deferred payment to suppliers, transportation routes, shipments, and more. The solutions allow the company to choose optimal positioning variant of a warehouse network [2]. Warehouse allocation optimization is a common task so there have been a number of articles reporting and proposing methods of solving it using linear programming. Three of them have been intriguing and gave some insights to our project.

An article published in 2019 [1] by Geomatics, "Natural Hazards and Risk" shows the optimization could be even used in building supply warehouses for disaster relief in China. The old system was not optimal, so they wanted to build a new model. The objective is to build several supply warehouses to delivery essential supplies for disaster relief in the region of Beijing, Tianjin, and Hebei with the cost minimized and lowest number of buildings constructed possible. Supply warehouses have different types (prefectural, county, and provincial level) and they have a certain number of demand points that these warehouses need to supply. The model is built similar to our project with the additional objective of minimum cost. The difference is they do not have a building type constraint, instead they have to build as few as possible while satisfying the objective of supplying all the demand points. The authors even show four optimal solutions but choose one to be the best with suggests building only county level disaster supply warehouses.

Another model proposed by Jia Shu in the article "An efficient greedy heuristic for warehouse-retailer network design optimization". The model considers four types of cost: transportation, inventory holding, fixed annual operation, and fixed ordering cost. The goal is to determine the optimal number of warehouses, allocation of retailers to warehouses, and optimal inventory replenishment strategies with the minimum cost of everything [3]. A mathematical formulation and an algorithm are proposed using 10 notations and 3 decision variables then run computationally. In order to fully run this model, a strong computer is required lest the model run for a very long time. This model proposed Pareto solutions and it has sophisticated mathematical formulations with even measured performance and time to run the model. The results yielded show interesting data: to choose from 200 instances, from building 1000 warehouses to 5000 warehouses with their corresponding costs so that the decision can be chosen more wisely.

Another research paper reviewed named "A Bi-objective Stochastic Optimization Model for Reliable Warehouse Network Redesign" proposes a model focused on a single product, single period, and two echelon logistics network comprising production, transshipment, and customer zone [4]. The objective is to minimize cost and maximize customer coverage. However, this model's scope is the biggest in all the articles found because it has 5 indices, 5 variables, and 17 parameters to consider from cost per unit capacity in warehouse, demand of customers for a scenario to probability of a scenario. Thus a single optimal solution cannot be found outright. This research has the best, most fleshed out, model so far with so many possibilities to look at with results in long and complicated mathematical formulations. It is only feasible with a Pareto optimal solution and depends on what the scenario company places their priorities on.

There are many optimization models out there, from simple to complex, to create an optimal warehouse network to fulfill the objective of minimum cost and maximum customers served. Depends on the criteria, scenario, and data that an appropriate model should be built to bring the best benefits to the companies.

4 Project Details and Methodology

A previous student of ETM 540, Auro, is now working for an international company headquatered in Seattle, Washington. This company is interested in building warehouses across the US, and would like to optimize their warehouse placement based on a number of factors. We have not been provided with the actual specifics, for privacy, and the possiblility that the details are not finalized yet. Auro has provided a Google Sheet that includes all of the constraints he was given. Our first step was to translate this spreadsheet into a mathematical model. Once that's complete, the next step is to use the mathematical model to write the R model. Below is a summary of the data elements provided, and the variables we chose to utilize.

We are given 7 distinct data elements:

- The number of Subregions: NSubregions
- – In the example provided, we have 15 subregions
- The number of Building Types: NBuildingTypes
- – In the example provided, we have 5 building types
- A the initial (starting) capacity (or 'delta') for each subregion
- B is the Building type limits; the total number of a given building type that can be built across all subregions.
- C is the Capacity of each building type
- R is the Regional Building Limits
- S is the Subregion building limits
- – Note that the subregional building limits are not guaranteed to be in alignment with the regional building limits.

Our model also has variables:

- x is the number of buildings, for each subregion, i, and each building type, j.
- If we were limited to one building of a given type per subregion, x could instead be a binary variable. However, there can be more than one of a given building type in any given subregion and thus x is an integer variable.
- An example of this variable is $x_{1,2} = 3$, which indicates there are 3 buildings of type 2 in subregion 1.
- L is effectively the lowest final capacity.
- U is effectively the highest final capacity.

We have three separate goals: raise the capacity floor, lower the capacity ceiling, and maximize total capacity. We want to focus on subregions with the lowest starting capacities and avoid putting too many building in a single subregion. We can prioritize these different goals by using the weights W_T, W_L, and W_U.

The mathematical formulation of the LP model we are solving is

$$\text{Maximize } W_T \sum_{i=1}^{15} \sum_{j=1}^{5} C_j x_{i,j} + W_L L - W_U U$$

$$\begin{aligned}
\text{s.t. } \sum_{i=1}^{4} \sum_{j=1}^{5} x_{i,j} &\leq R_1, \\
\sum_{i=5}^{7} \sum_{j=1}^{5} x_{i,j} &\leq R_2, \\
\sum_{i=8}^{12} \sum_{j=1}^{5} x_{i,j} &\leq R_3, \\
\sum_{i=13}^{13} \sum_{j=1}^{5} x_{i,j} &\leq R_4, \\
\sum_{i=14}^{15} \sum_{j=1}^{5} x_{i,j} &\leq R_5, \\
\sum_{j=1}^{5} x_{i,j} &\leq S_i \ \forall\, i \\
\sum_{i=1}^{15} x_{i,j} &\leq B_j \ \forall\, j \\
\sum_{j=1}^{5} C_j x_{i,j} + A_i &\geq L \ \forall\, i \\
\sum_{j=1}^{5} C_j x_{i,j} + A_i &\leq U \ \forall\, i
\end{aligned}$$

5 R Model and Results

This model is fully generalizable, so changes in the following code chunk can be used to analyze different constraints. It's common that these kinds of problems don't have set numbers for very long. The obvious, first setting for the weights would be to set them all to 1.

```
#WEIGHTS
W_L <- 1 #Maximizing the lowest capacities (focus on raising the minimum)
W_U <- 1 #Minimizing the highest capacities (avoid high capacity outliers)
W_T <- 1 #Maximize the total capacity

#SYSTEM ATTRIBUTES
NSubregions <- 15 #Number of subregions
NBuildingTypes <- 5 #Number of building types

StartingCapacities <- c(-20,15,2,5,-1,4,-3,20,4,-5,8,-3,2,1,4)
```

```r
S <- c(2,3,4,3,
       3,2,3,
       2,5,6,5,5,
       2,
       2,3) #Sub-region Limits

R <- c(5,10,1,2,4) #Region Limits

B <- c(4,3,2,4,2) #Building Type LImits

C <- c(20,16,14,10,25) #Capacity per Building Type
```

The model implementation in R can be seen in the following code chunk.

```r
model <- MIPModel()
model <- add_variable(model, vx[i,j], type = "integer",
                      i=1:NSubregions,
                      j=1:NBuildingTypes, lb=0)
model <- add_variable(model, vL, type = "integer")
model <- add_variable(model, vU, type = "integer")

model <- set_objective(model, W_T*sum_expr(C[j]*vx[i,j],
                                           i=1:NSubregions,
                                           j=1:NBuildingTypes)+W_L*vL-W_U*vU
                       , "max")

#Limit buildings per region
model <- add_constraint(model, sum_expr(vx[i,j],i=1:4,j=1:5) <= R[1])
model <- add_constraint(model, sum_expr(vx[i,j],i=5:7,j=1:5) <= R[2])
model <- add_constraint(model, sum_expr(vx[i,j],i=8:12,j=1:5) <= R[3])
model <- add_constraint(model, sum_expr(vx[13,j],j=1:5) <= R[4])
model <- add_constraint(model, sum_expr(vx[i,j],i=14:15,j=1:5) <= R[5])


#Limit number of buildings per sub-region
model <- add_constraint(model,sum_expr(vx[i,j], j=1:NBuildingTypes) <= S[i],
                        i=1:NSubregions)

#Limit number of building types
model <- add_constraint(model,sum_expr(vx[i,j], i=1:NSubregions) <= B[j],
                        j=1:NBuildingTypes)

#Lower bound must be below all subregion final capacities
model <- add_constraint(model,sum_expr(C[j]*vx[i,j],
                                       j=1:NBuildingTypes)>=vL-StartingCapacities[i],
                        i=1:NSubregions)

#Upper bound must be above all subregion final capacities
model <- add_constraint(model,sum_expr(C[j]*vx[i,j],
                                       j=1:NBuildingTypes)<=vU-StartingCapacities[i],
                        i=1:NSubregions)

modelSol <- solve_model(model, with_ROI(solver = "glpk"))
```

```r
results <- get_solution (modelSol, vx[i,j])
resultstable <- matrix(results[,4],ncol=5)

finalL <- get_solution(modelSol, vL)
finalU <- get_solution(modelSol, vU)
```

Now, the results of the model with all of the weights being equal can be seen below:

```
## Mixed integer linear optimization problem
## Variables:
##   Continuous: 0
##   Integer: 77
##   Binary: 0
## Model sense: maximize
## Constraints: 55
```

```
## vL
## -3
```

```
## vU
## 25
```

```
##       [,1] [,2] [,3] [,4] [,5]
##  [1,]    0    1    0    0    1
##  [2,]    0    0    0    1    0
##  [3,]    0    0    1    0    0
##  [4,]    1    0    0    0    0
##  [5,]    0    1    0    1    0
##  [6,]    1    0    0    0    0
##  [7,]    0    1    0    1    0
##  [8,]    0    0    0    0    0
##  [9,]    0    0    0    0    0
## [10,]    0    0    0    0    1
## [11,]    0    0    0    0    0
## [12,]    0    0    0    0    0
## [13,]    1    0    0    0    0
## [14,]    0    0    1    1    0
## [15,]    1    0    0    0    0
```

StartingCapacities	FinalCapacities
−20	21
15	25
2	16
5	25
−1	25
4	24
−3	23
20	20
4	4
−5	20
8	8
−3	−3
2	22
1	25
4	24

As we can see, with the given constraints, we still weren't able to get all of the subregions to positive final capacities. With that being the case, there is no real point in further prioritizing L. Instead, let's increase W_U to 5 and see the results.

```
#New Weights
W_L <- 1
W_U <- 5
W_T <- 1
```

```
## Mixed integer linear optimization problem
## Variables:
##   Continuous: 0
##   Integer: 77
##   Binary: 0
## Model sense: maximize
## Constraints: 55
```

```
## vL
## -3
```

```
## vU
## 25
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    1    0    1
```

```
##  [2,]    0    0    0    1    0
##  [3,]    1    0    0    0    0
##  [4,]    1    0    0    0    0
##  [5,]    0    1    0    1    0
##  [6,]    0    1    0    0    0
##  [7,]    0    1    0    1    0
##  [8,]    0    0    0    0    0
##  [9,]    0    0    0    0    0
## [10,]    0    0    0    0    1
## [11,]    0    0    0    0    0
## [12,]    0    0    0    0    0
## [13,]    1    0    0    0    0
## [14,]    0    0    1    1    0
## [15,]    1    0    0    0    0
```

StartingCapacities	FinalCapacities
−20	19
15	25
2	22
5	25
−1	25
4	20
−3	23
20	20
4	4
−5	20
8	8
−3	−3
2	22
1	25
4	24

This time, we still had the same total capacity overall, but we evened out the capacity allocation between subregions 1, 3, and 6. This result clearly meets the expectations of our three goals.

6 Model Discussion & Future Adaptations

With the example data we were given, we were able to find a solution which met all of the goals laid out by Auro. However, the real data could look slightly different. With additional time, there could be added value in looking at the sensitivity of the three weights we have. Due to the fact that the total additional capacity will almost always be notably larger than the highest capacity and smallest capacity, the weight W_T will have naturally have a greater impact on the outcome of the model. So, when we set each of the weights to be equal to 1, that didn't necessarily mean that each of the goals were being considered equally as important. Furthermore, we were told that cost was not a factor in this project. While that helped simplify the model, it likely would not be applicable to most companies. Future adaptations to this model most certainly should allow for the inclusion of cost. Let's take a brief look into how it might be implemented. Firstly, we would need a cost matrix that represents the cost of a certain building type in a given subregion. Perhaps there may need to be a third dimension to the matrix which allows for two separate buildings of the same type in the same subregion to cost different amounts. Just because the building type is the same and it's in the same subregions would not necessarily mean that the buildings will cost the exact same amount. Once we had that data, our model could include a constraint which limits total spending if a company has a budget they're working with. Alternatively, they might not have a strict budget and are simply concerned with cost efficiency. In both cases, we would want to include total cost in our objective function. We could also add a weight for total cost as well so that we can allow for the same prioritization of objectives as we currently have with capacity. There is certainly space for this model to expand but we believe that, for the purposes of this project, this model can act as a useful tool to make intelligent decisions about where and what to build.

7 Conclusion

We were able to build a simple linear program to maximize capacity across all regions and building types. Moreover, our model is a simplified representation of a real world scenario of resource allocation and data warehousing by design from scrubbed data provided. As further development on our initial solution, we identified a weak point and constructed a more appropriate model using the same goals and constraints.

As a solution development after we did identify or construct an appropriate model that there was given the goals and constraints we had been provided. We were able to build a simple linear program to maximize capacity across all regions and building types. Moreover, our model is the simplified representation of the real world such as a resource allocation and data warehousing. We did development of #management scheme, and the model represented an optimal solution.

However, there was not just one goal. So, we modified our original model to set weights for raising the capacity floor, lowering the ceiling, and maximizing overall capacity. This allowed us to discover different outcomes, all of which could be considered optimal in their own way: one with capacity more evenly distributed throughout subregions and the other less so. There are site areas for having the zone of a truck which becomes a green zone. The technique of a truck free zone and having a warehousing just outside the city and it becomes a green loop. This technique can be one of the future solutions. Furthermore, besides cost, we can modify the model if the company desires to look for warehouse locations that have the shortest distance to points of interest. We can add a set of distance data, a new variable for distance and add an objective for minimum distance. However, this might result in possible Pareto solutions since it is difficult to satisfy several objectives simultaneously. Finally, not only were we able to find optimal solutions for the given set of constraints, but the model was also generalizable so that it could be used as a tool with different numbers.

8 References

1. Ma, Y., Xu, W., Qin, L., Zhao, X., & Du, J. (2019). Hierarchical supplement location-allocation optimization for disaster supply warehouses in the Beijing–Tianjin–Hebei region of china. Geomatics, Natural Hazards & Risk, 10(1).

2. “Supply Chain Optimization: Choosing the Right Location for Warehouses.” – AnyLogic Simulation Software, AnyLogic Company, www.anylogic.com/warehouse-network-development/

3. Shu, Jia. “An Efficient Greedy Heuristic for Warehouse-Retailer Network Design Optimization.” Transportation Science, vol. 44, no. 2, 2010, pp. 183–192. JSTOR.

4. Afar Razmi, AmirHossien Zahedi-Anaraki, MohammadSaleh Zakerinia (2013). A bi-objective stochastic optimization model for reliable warehouse network redesign, Mathematical and Computer Modelling, Volume 58, Issues 11–12,2013, Pages 1804-1813, ISSN 0895-7177.