2017

Algorithmic Methods for Concave Optimization Problems with Binary Variables

Tao Li
Lehigh University

Algorithmic Methods for Concave Optimization Problems with Binary Variables

by

Tao Li

Master of Science Degree

Presented to the Graduate and Research Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial and Systems Engineering

Lehigh University

December 8, 2016

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Master of Science.

____

Date

____

Lawrence Snyder, Thesis Advisor

____

Tamás Terlaky, Chairperson of Department

Acknowledgements

I want to thank my parents for the supporting throughout my life and study. Also, I'd like to thank Professor Snyder for his patience, encouragement and guidance in the master's thesis process.

Contents

List of Tables

List of Figures

In this thesis, we discuss two problems that involve concave minimization problems with binary variables.

For the first problem, we compare the Lagrangian relaxation method and our reformulation method in solving the location model with risk pooling (LMRP) with constant customer demand rate and equal standard deviation of daily demand.

Our new method is to reformulate the original non-linear model for the LMRP as a linear one. In the reformulation, we introduce a set of parameters representing the increase in the cost of the non-linear part for each additional cost assigned to a potential facility site, and a set of new decision variables indicating how many customers are assigned to each facility. Then we get a linear model by replacing the none-linear part of the original objective function by the sum of the additional costs from 0 to the number of customers assigned.

The algorithms are tested on problems with 5 to 500 potential facilities and randomly generated locations. The computation time of the Lagrangian method grows slower as the number of potential sites increases than the linear method does. Though the new method takes more time than the Lagrangian method, it may have better efficiency for solving problems with special patterns in the distance matrix or other special structure.

In the second half of this thesis, we discuss a general multiple-square-root pricing problem. We test a heuristic that involves sorting, similar to the method for solving the Lagrangian subproblem for the LMRP, and we explore conditions under which the heuristic could be optimal. The accuracy rates for this method are decreasing at a slow rate as the number of the square root terms grows, while it stays high when the number is not too big.

Chapter 1

Introduction

1.1 The Location Model with Risk Pooling

The location model with risk pooling (LMRP) is introduced by [3]. This problem chooses facility locations in order to minimize the total cost of building the facilities, transporting goods from facilities to customers and holding inventory to take advantage of economies of scale and protect against uncertain demand. So there are four parts in the objective function; the construction cost, the transportation cost, the cycle stock cost and the safety stock cost. This problem can be solved quite efficiently using Lagrangian relaxation when the ratio of the demand variance to mean is the same for every customer.In our thesis, we consider a further special case in which we assume all facilities share the same customer demand rate and standard deviation of daily demand.

Our work was motivated by the modeling approach used in the Maximum Expected Covering Location Problem (MEXCLP). MEXCLP is introduced by [2]. This model is a kind of covering problem; it decides the number of vehicles in each location in order to maximize the expected number of demands that can be covered, given that vehicles may be unavailable (in use). The model assumes that there is an equal probability that a vehicle is busy at any location. As the objective function is the expected number of demands, the decision variables that choose the number of vehicles in each location appear in an exponential term. This makes the objective function non-linear, just like the LMRP problem. Daskin introduces a set of parameters to represent the increase in the expected coverage for each additional vehicle, as well as a set of binary decision variables to indicate whether the customer is covered a specific numbers of times. By using the sum of all the benefits of adding a new vehicle to represent the expected coverage, he changes the problem into one that is linear and easy to solve. So we apply the same idea to convert the LMRP into a linear mixed-integer programming problem and compare it with the Lagrangian method of [3] to see if it will give us a more efficient method.

1.2 Multiple-Square-Root Minimization Problem

The Multiple-Square-Root Minimization Problem (MSR) has an objective functions that consists of a sum of a linear term and at least two square root terms. The Lagrangian sub-problem for the LMRP is a typical MSR problem, and there are other MSR problem in real life. A simple example is that we add other concave cost besides the safety stock cost to the LMRP, such as the labor cost and even minimize the negation of the revenue. To solve the Lagrangian sub-problem for the LMRP problem, [3] and [6] assumes the ratio of the demand variance to mean is the same for every customer and collapse the two square root terms into one, then use a sorting algorithm to solve the resulting sub-problem. Our focus is on the two-square-root problem in which this assumption does not hold. We address this problem along two directions. The first is to relax the assumption in the LMRP's Lagrangian method and the second one is to add a condition for the parameters that makes the sorting method always optimal.

The remainder of this thesis is organized as following. In Chapter 2 we review relevant related literature. The reformulated linear model is described in Chapter 3. Chapter 4 outlines the computational results and a comparison of solution times for the linear and Lagrangian methods. The Multiple-square-root Minimization problem is analysed in Chapter 5. In Chapter 6, we present our conclusions and outline directions for future work.

Chapter 2

Literature Review

In 2002 [3] introduced the LMRP(Location Model with Risk Pooling). The original motivation of the model was a study of a Chicago-area blood bank system. Compared with the UFLP model, the LMRP takes cycle stock and safety stock into account. So there are two more non-linear terms in the objective function. [3] then proposed a Lagrangian method to solve this model for a special case in which the ratio of the demand variance to the demand mean is identical for all retailers. Then in 2003, Shen et al [6] presented a different algorithm for solving the LMRP. They reformulated the model into a set covering problem and applied a column generation algorithm to solve it. The computational results shows that this algorithm is fast, but still slower than the Lagrangian method. Snyder, Daskin and Teo [8] presented a stochastic version of the LMRP problem, and they developed a Lagrangian method for this problem. They also discussed the influence of changing the key parameters. Ozsen [4] considered the interdependence between capacity and inventory management in the LMRP. The Lagrangian sub-problem is also a non-linear integer program. They proposed an efficient algorithm for the continuous relaxation of this sub-problem.

The Maximum Expected Covering Location Problem (MEXCLP) is introduced by Daskin [1] and [2]. The MEXCLP chooses locations of facilities that can sometimes be unavailable and Daskin artfully replaces the none-linear term in the objective function by a sum of linear terms. That linearizes the model and makes it easier to solve. In our thesis we apply this method to the LMRP problem.

A couple of special cases of the Multiple-Square-Root Minimization problem are presented by Shen et al [5]. They listed problems like [7] and [6] which all had at least two concave (square root) terms in the objective function. Shu developed an efficient algorithm to solve this kind of problem in [7] through a corresponding concave minimization problem defined on a polyhedron. In our thesis, we try to find under what condition the sorting algorithm for the LMRP Lagrangian sub-problem can be applied to the general two square root pricing problem.

Chapter 3

Model Formulation

3.1 Maximum Expected Covering Location Problem

As we mentioned in the Introduction chapter, our approach for solving concave binary minimization problems is inspired by a reformulation strategy that is sometimes used to solve other binary optimization problems in which the objective function contains a non-linear function of the sum of the binary variables. The basic idea is to introduce auxiliary parameters and binary variables and use their product to represent the none-linear part, and use these to linearize the objective function.

One model that uses this approach is the maximum expected covering loction problem (MEXCLP) by [2]. The MEXCLP chooses locations of facilities that can sometimes be unavailable (e.g., because the ambulance located there is busy on another call). A demand node is covered by a facility if it is within a certain coverage radius of it. The goal of the MEXCLP is to locate at most P facilities to maximize the total expected coverage of the demand nodes.

The MEXCLP assumes that the probability that a facility is unavailable at any time is given by q. It also assumes that facility unavailabilities are independent, so if there are n facilities that cover a demand node, then the probability that all of them are unavailable is given by q^n. Since the number of covering facilities, n, is not known a priori, we have to express it in terms of the decision variables as $\sum_{j \in J} a_{ij} X_j$, where a_{ij} is a parameter that equals 1 if facility j covers demand node i and 0 otherwise. Then the model can be formulated as follows:

Parameters:

J set of potential facilities, indexed by j,

I set of customer nodes, indexed by i,

q the probability that a facility is unavailable at any time,

P the maximum number of the facilities can be chosen,

h_i the demand generated at node i,

$$a_{ij} = \begin{cases} 1 & \text{if a facility at } j \text{ can cover demands at customer node } i, \\ 0 & \text{otherwise,} \end{cases}$$

Decision Variables:

X_j the number of facilities to be built at j

Then the model can be formulated as follows:

$$\text{Maximize} \quad \sum_{i \in I} h_i \left(1 - q^{\sum_{j \in J} a_{ij} X_j}\right)$$

$$\text{subject to} \quad \sum_{j \in J} X_j \leq P$$

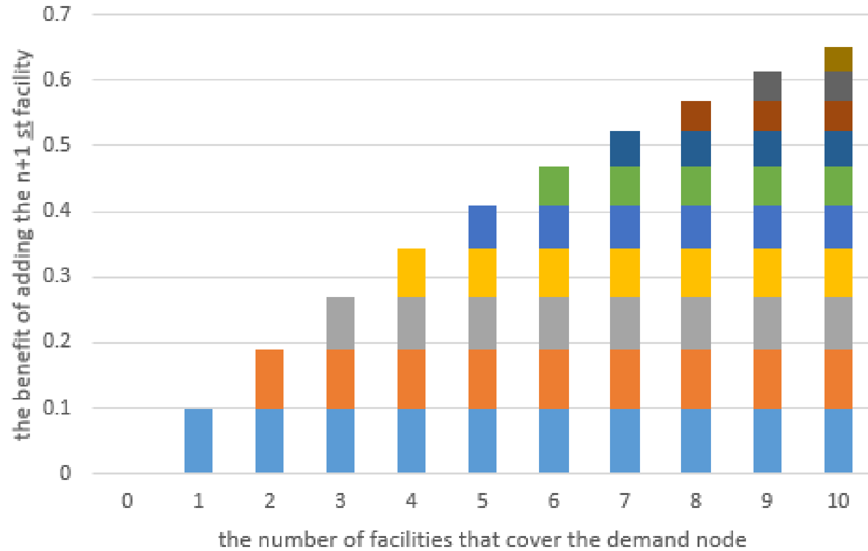
$$X_j \in 0, 1 \quad \forall j \in J$$

3.2 Linearization of Maximum Expected Covering Location Problem

In the original formulation, the probability that the demand of a customer node i is covered is given by $1 - q^{\sum_{j \in J} a_{ij} X_j}$, which is a non-linear function of X_j. Instead of computing the probability directly, [2] proposes adding up the benefits of each new facility. We now summarize his approach.

Figure (3.1) shows how we compute the probability by adding up benefits. The x-axis is the number of facilities that cover the demand node and the y-axis is the probability. Assume there are k facilities that can cover the demand node. We first compute the benefit of adding the

Figure 3.1: Adding up benefits to compute probability



$(n+1)$st facility (assuming we already have n facilities) $(n = 0, 1, ..., P-1)$ and then we can add them up from $n = 1$ to k.

The availability probability for n facilities is $1 - q^n$. Thus the benefit of adding another source to cover the node from n facilities is $(1 - q^{n+1}) - (1 - q^n) = q^n(1 - q)$. We introduce a new variable Z_{jk} to represent the number of times covered, which we define to be 1 if demand node i is covered k or more times, and 0 if not. The model then can be formulated as follows.

$$\text{Maximize} \quad \sum_{k=1}^{P} \sum_{i \in I} h_i q^{k-1}(1-q) Z_{jk}$$

$$\text{subject to} \quad \sum_{j \in J} a_{ij} X_j - \sum_{k=1}^{P} Z_{ik} \geq 0 \quad \forall i \in I$$

$$\sum_{j \in J} X_j \leq P$$

$$X_j \in 0, 1 \quad \forall j \in J$$

$$Z_{jk} \in 0, 1 \quad \forall j \in J; k = 1, 2, ..., P.$$

In this model, we add up the benefits to replace the none-linear part of the objective function and that gives us a linear formulation. In what follows we propose a similar method to reformulate the LMRP model as a linear one.

3.3 The Location Model with Risk Pooling

The LMRP model is an extension of the UFLP that considers uncertain demand. Besides the fixed cost of opening locations and the variable transportation cost, it also includes the cost of cycle stock and safety stock. As a result, the LMRP is structured much like the UFLP model, with two extra non-linear terms in the objective function. Despite its concave objective function, the LMRP problem can be solved by Lagrangian relaxation quite efficiently, just like the UFLP, assuming that the ratio of the customer demand rate μ and the standard deviation of daily demand σ are constant. We use the following notations:

Parameters:

I set of retailers, indexed by i,

J set of candidate DC sites, indexed by j,

μ_i mean daily demand of retailer i, for each $i \in I$,

σ_i^2 variance of daily demand of retailer i, for each $i \in I$,

f_j fixed (daily) cost of locating a DC at candidate site j, for each $j \in J$,

K_j fixed cost for DC j to place an order from the supplier, including fixed components of both ordering and transportation costs, for each $j \in J$,

d_{ij} cost per unit to ship between retailer i and candidate DC site j, for each $i \in I$ and $j \in J$.

Θ a constant parameter that captures the safety stock costs at candidate sites.

Decision Variables:

$$X_j = \begin{cases} 1 & \text{if we locate at candidate site } j, \\ 0 & \text{if not,} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 & \text{if demands at retailer } i \text{ are assigned to a DC at candidate site } j, \\ 0 & \text{if not.} \end{cases}$$

Then the model is formulated as follows.

$$\text{Minimize} \quad \sum_{j\in J}\left\{f_jX_j+\sum_{i\in I}d_{ij}Y_{ij}+K_j\sqrt{\sum_{i\in I}\mu_iY_{ij}}+\Theta\sqrt{\sum_{i\in I}\sigma_i^2Y_{ij}}\right\}$$

$$\text{subject to} \quad \sum_{j\in J}Y_{ij}=1 \quad \forall i\in I$$

$$Y_{ij}\le X_j \quad \forall i\in I,\forall j\in J$$

$$X_j\in\{0,1\}, \quad \forall j\in J$$

$$Y_{ij}\in\{0,1\}, \quad \forall i\in I,\forall j\in J.$$

3.4 Linearization of LMRP

To make the objective function linear, we introduce a new parameter γ_{jk} to represent the cost of safety and cycle stock cost that k retailers are assigned to DC j, that is

$$\gamma_{jk}=K_j\sqrt{k\mu}+\Theta\sqrt{k\sigma^2}.$$

Also we introduce a new decision variable

$$Z_{jk}=\begin{cases}1, \text{ if exactly } k \text{ retailers are assigned to DC } j,\\ 0, \text{ if not}\end{cases}$$

To associate Z_{jk} with its meaning using linear constraints, we add the constraints

$$\sum_{k=0}^{|I|}kZ_{jk}=\sum_{i\in I}Y_{ij} \quad \forall j\in J$$

$$\sum_{k=0}^{|I|}Z_{jk}=1 \quad \forall j\in J.$$

The second constraint says that only one of the Z_{jk} can be equal to 1 for each j and the first constraint makes sure that the 1 appears when $k=\sum_{i\in I}Y_{ij}$, which is just how we define the meaning of Z_{jk}.

So the linear model is:

$$\text{Minimize} \quad \sum_{j\in J}\left\{f_jX_j + \sum_{i\in I} d_{ij}Y_{ij} + \sum_{k\in J}\gamma_{jk}Z_{jk}\right\} \tag{3.1}$$

$$\text{subject to} \quad \sum_{j\in J} Y_{ij} = 1 \quad \forall i \in I \tag{3.2}$$

$$Y_{ij} \le X_j \quad \forall i \in I, \forall j \in J \tag{3.3}$$

$$\sum_{k=0}^{|I|} kZ_{jk} = \sum_{i\in I} Y_{ij} \quad \forall j \in J \tag{3.4}$$

$$\sum_{k=0}^{|I|} Z_{jk} = 1 \quad \forall j \in J \tag{3.5}$$

$$X_j \in \{0,1\}, \quad \forall j \in J \tag{3.6}$$

$$Y_{ij} \in \{0,1\}, \quad \forall i \in I, \forall j \in J \tag{3.7}$$

$$Z_{jk} \in \{0,1\}, \quad \forall j \in J, \forall k = 0, ..., |I|. \tag{3.8}$$

From these two formulations, we can see although the second method is linear,it has many more constraints than the original formulation. On the other hand, it can be solved by an off-the-shelf MIP solver and does not require Lagrangian relaxation as in the original LMRP. So it's hard to say which computation time would be shorter only by looking at the models. We will test randomly generated examples and compare the solution time of the two methods in Chapter 4.

3.5 The Lagrangian Relaxation Method for the LMRP

Similar to the UFLP, we solve the LMRP by relaxing the assignment constraints equation (3.2) to obtain the following Lagrangian sub-problem:

$$\text{Minimize} \quad \sum_{j\in J}\left\{f_jX_j + \sum_{i\in I} d_{ij}Y_{ij} + K_j\sqrt{\sum_{i\in I}\mu_iY_{ij}} + \Theta\sqrt{\sum_{i\in I}\sigma_i^2Y_{ij}}\right\} + \sum_{i\in I}\lambda_i\left(1 - \sum_{j\in J}Y_{ij}\right)$$

$$= \text{Minimize} \quad \sum_{j \in J} \left\{ f_j X_j + \sum_{i \in I} (d_{ij} - \lambda_i) Y_{ij} + K_j \sqrt{\sum_{i \in I} \mu_i Y_{ij}} + \Theta \sqrt{\sum_{i \in I} \sigma_i^2 Y_{ij}} \right\} + \sum_{i \in I} \lambda_i$$

$$\text{subject to} \quad Y_{ij} \leq X_j \quad \forall i \in I, \forall j \in J$$

$$X_j \in \{0, 1\}, \quad \forall j \in J$$

$$Y_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J.$$

Although the sub-problem is a concave integer minimization problem, it can be solved relatively efficiently, using a sorting method developed by [3] and [6]. The algorithm relies on the assumption that the ratio of the demand variance to the demand mean is a constant for all retailers. That is, for all $i \in I$, $\sigma_i^2/\mu_i = \gamma$ for some constant $\gamma \leq 0$. Then we can collapse two square root terms into one and apply the sorting algorithm to solve the resulting sub-problem. In the multiple square root pricing problem, we focus on the two-square-root problem in which this assumption does not hold.

The optimal objective function value of the Lagrangian sub-problem gives us a lower bound of the original problem; then we need an upper bound. There are many ways to find a feasible solution to get the upper bound; in this paper, we use a simple algorithm to generate the solution from the sub-problem result. This is shown in the appendix.

Finally, we recursively update λ to get a smaller gap between the lower and upper bound. Our stopping condition in the computational tests in this thesis is when the number of iterations is over 500 or the gap is less than or equal to 5 percent of the upper bound. There is no limit for cpu time since the first stopping condition includes it.

Chapter 4

Computational Results

4.1 Test on Random Instances

In this section, we outline computational results from 10 experiments with $|I|$ and $|J|$ ranging from 5 to 500 in increments of 5. Tables (4.4) and (4.5) show a sample of our random data set with data scale of 5. Our data set has $|I| = |J|$ and we define data scale as $|I|$ (the number of retailers). The parameters are generated randomly in Excel and the distributions are shown in Table 4.2. We also saved the upper bound and the lower bound at each iteration during the Lagrangian method. This shows the interval change of each iteration and helps us choose a suitable stopping condition.

We implemented the Lagrangian method in C++ and the linearization method in AMPL with CPLEX version 12.4.0.0. Table (4.1) is the comparison of the solution time for the linearization and Lagrangian methods. The linearization method has a similar solution time as the Lagrangian method when the number of facilities is small. However, the solution time increases faster with the number of facilities for the linearized method than it does for the Lagrangian method. So for larger scale problems (which are more practical) the Lagrangian method will have better performance.

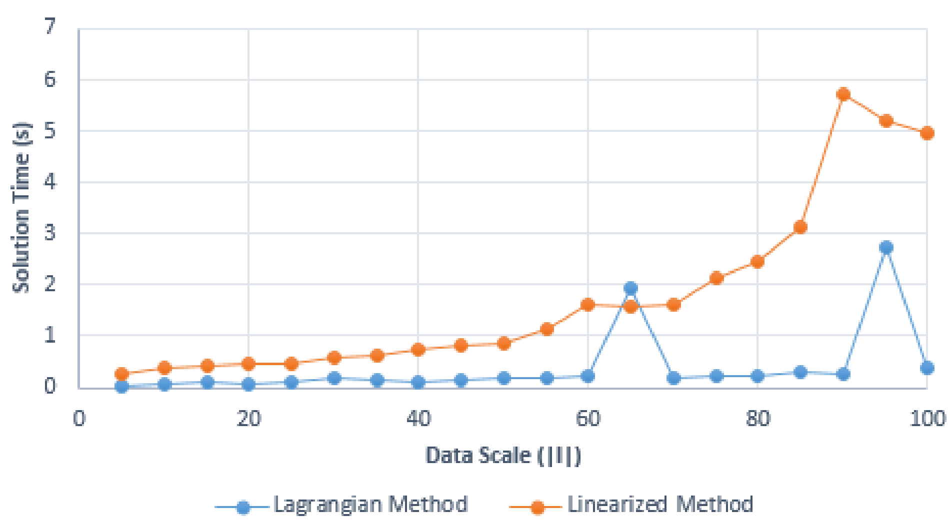
In our experiments, for one specific data set, CPLEX gets stuck when it tries to solve the linearization problem. It takes over 90 seconds while the other samples with the same data scale only need 2.91 seconds on average. When we use the Lagrangian method to solve the same problem, the method also stopped because the number of iterations is over 500. The reason that the Lagrangian method can't solve this kind of data set in a small number of iterations is that the Lagrangian relaxation's optimal value can't reach the original problem's optimal value, and the gap is over 5 percent, but the reason why the gaps are large for this data set is not clear. Similarly, we still can't understand why CPLEX also gets stuck for this data set.

The gap between the upper bound and the lower bound is not large; it is on average 4.3 percent. However, there is a significant increase in the gap when the data scale grows larger.

Figure (4.2), Figure (4.3) and Table (4.1) are the comparison of the average solution time for the Lagrangian method and the linearization method. In the figure, the lower line (blue) represents the solution time for the Lagrangian method and the upper line (orange) is for the linearization method.

Figure (4.4), Figure (4.5) and Table (4.2) are the comparison of the Maximum solution time for the Lagrangian method and the linearization method.

Figure 4.1: Average solution time for data scale 5 to 100



Data Scale	Lagrangian Method	Linearized Method
5	0.039	0.264
10	0.055	0.373
15	0.105	0.422
20	0.081	0.447
25	0.096	0.472
30	0.171	0.567
35	0.133	0.623
40	0.109	0.729
45	0.153	0.815
50	0.175	0.873
55	0.169	1.151
60	0.213	1.626
65	1.948	1.559
70	0.172	1.597
75	0.225	2.142
80	0.240	2.458
85	0.289	3.121
90	0.246	5.722
95	20715	5.215
100	0.373	4.965
150	0.726	31.408
200	7.645	300.122
250	10.753	193.101
300	2.540	650.172
350	3.750	397.260
400	27.234	166.028
450	59.588	397.333
500	71.281	364.518

Table 4.1: Average solution time comparison

Based on the results, we can see that the gap between the lower and upper bounds is acceptable, even in the cases that stopped due to the 500-iterations limit. Additionally, when the scale is not large, we can get results from the Lagrangian method with a tiny gap, say 0.5 percent. So it can be concluded that the Lagrangian method is reliable.

From Table (4.1), the linearization method becomes slower than the Lagrangian method on average when the data scale is large. From Table (4.4), we see that the maximum solution time of the ten samples has a similar trend as the average solution time. However, there do exist instances for which the linearization method runs faster. The solve-speed depends on the instances;

Lagrangian Method	Linearized Method	
5	0.078	0.420
10	0.086	0.828
15	0.125	0.623
20	0.192	0.803
25	0.202	0.733
30	0.233	0.826
35	0.187	1.022
40	0.162	1.483
45	0.329	1.891
50	0.245	1.623
55	0.227	1.92
60	0.297	2.427
65	6.935	2.895
70	0.502	2.798
75	0.643	4.052
80	0.582	6.304
85	0.636	8.471
90	0.721	12.567
95	9.028	15.465
100	1.873	13.981
150	2.234	90.385
200	21.43	639.293
250	28.388	293.233
300	8.449	1243.54
350	14.284	1539.455
400	53.293	324.144
450	19.116	567.342
500	20.101	597.369

Table 4.2: Maximum solution time comparison

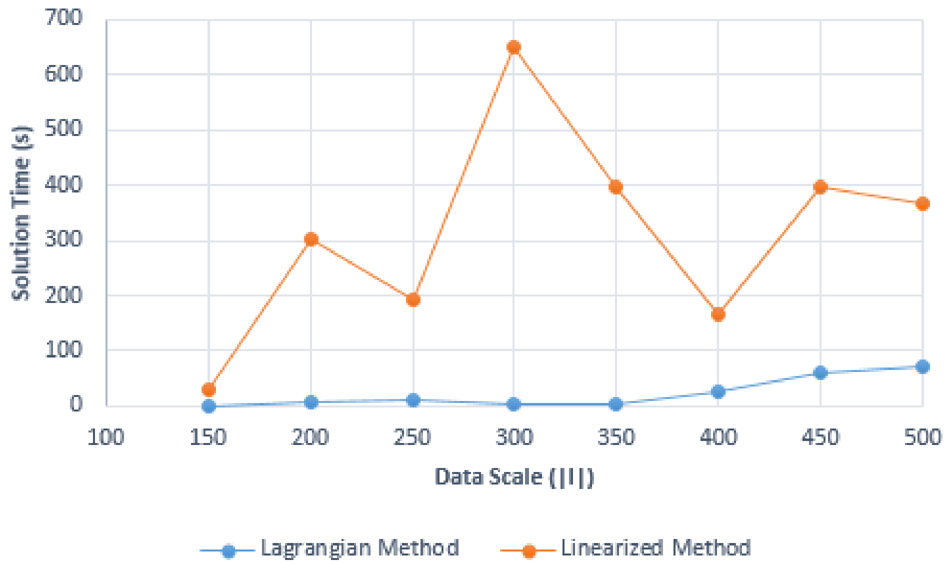
for further research, we will test on larger scale problems and real world instances.

4.2 Test on Benchmark Instances

As in real life, the data such as distance, fixed cost of opening a new facility and the demand of different places are not always independent, so it is necessary to compare the computation time not only on random data sets, but also on examples that come from more realistic instances.

Our data comes from [3] and we use two data sets. For the 88-node dataset, representing

Figure 4.2: Average solution time for data scale 150 to 450



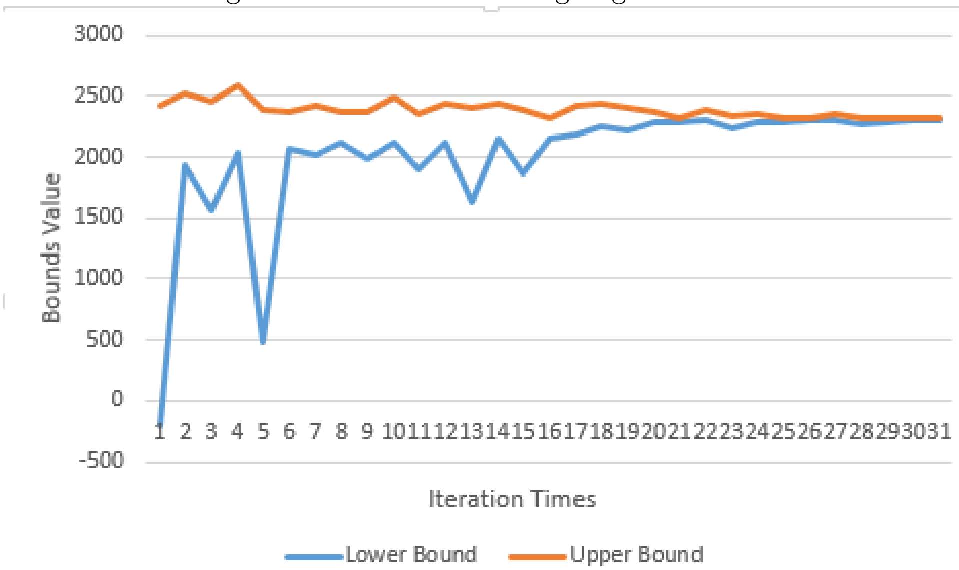
Parameters	Generated from distribution
d_{ij}	Uniform[0,500]
f_j	Uniform[0,15]
K_j	Uniform[0,25]
Θ	Inverse Normal
σ^2	Uniform[0,1]
μ	Uniform[0,2]

Table 4.3: Data distribution for LMRP

the 50 largest cities in the 1990 U.S. census along with the 48 capitals of the continental U.S. minos duplicates, the mean demand was obtained by dividing the population data by 1000 and rounding the result to the nearest integer. Fixed facility location costs were obtained by dividing the facility location costs by 100. For the 150-node dataset, representing the 150 largest cities in the continental U.S. for the 1990 census, the mean demand was obtained in the same manner. The fixed facility costs were all set to 100, one thousandth of the value in the dataset given by [3]. These changes were made to allow us to deal with smaller numbers.

For the 88-node dataset, the solution time for the Lagrangian method is 0.203 s and it takes CPLEX 2.435 s. For the 150-node dataset, the solution time for the Lagrangian method is 0.539 s and it takes CPLEX 19.673 s. We see that the solution time for both methods is a little bit smaller than the average of the random samples and the Lagrangian method is still much faster

Figure 4.3: Bounds from Lagrangian Method

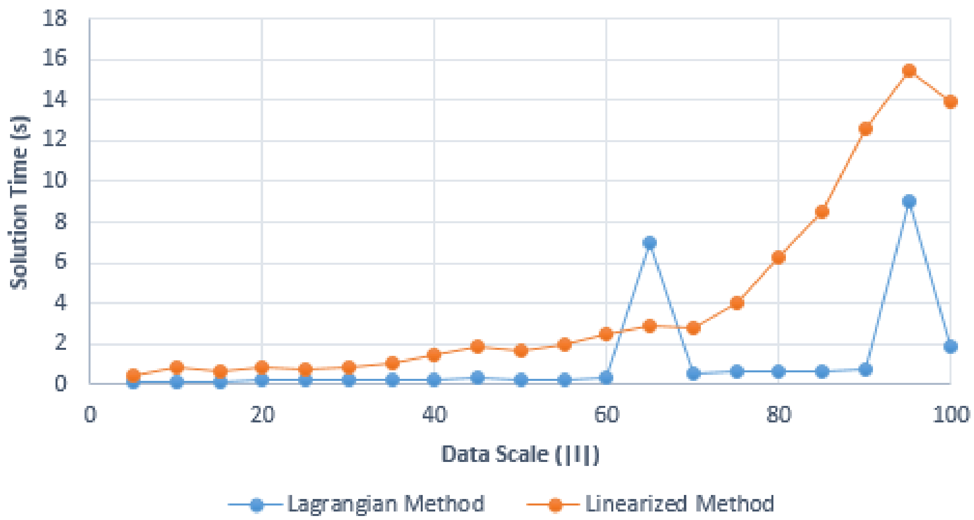


$j(i)$	f_j	K_j	Θ	μ	σ^2
1	6.7899	20.1892	-0.6551	0.1004	1.4484
2	8.3984	10.8181	-0.6551	0.1004	1.4484
3	5.4865	24.0214	-0.6551	0.1004	1.4484
4	3.2355	14.2425	-0.6551	0.1004	1.4484
5	8.7225	10.2614	-0.6551	0.1004	1.4484

Table 4.4: Sample test data for LMRP

than the linearization method. So the randomness of the initial instances may not have much influence on the comparison of these two methods.

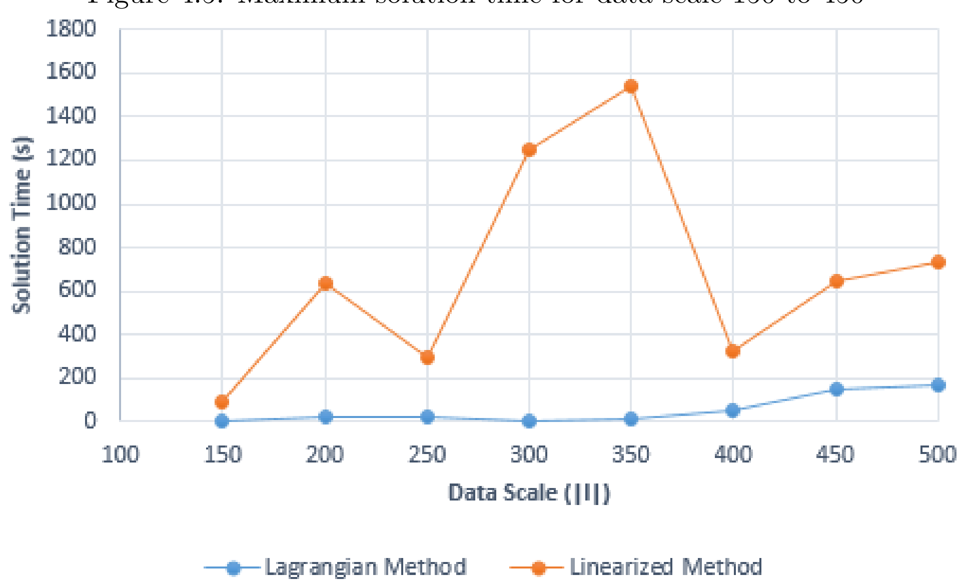
Figure 4.4: Maximum solution time for data scale 5 to 100



d_{ij}	1	2	3	4	5
1	313.0018	427.8025	148.3325	269.3765	47.4767
2	307.9506	275.2097	240.9874	236.6587	39.4540
3	210.9449	216.0247	295.6569	200.8956	55.5366
4	228.2597	245.9648	386.1843	321.6904	185.5226
5	83.5940	316.4902	476.5017	329.6392	48.3012

Table 4.5: Sample test data for LMRP

Figure 4.5: Maximum solution time for data scale 150 to 450



Chapter 5

Multiple Square Root Minimization Problem

5.1 Model Introduction

In the LMRP model, the Lagrangian sub-problem can be written as

$$\text{Minimize} \quad \sum_{j \in J} \left\{ f_j X_j + \sum_{i \in I} (d_{ij} - \lambda_i) Y_{ij} + K_j \sqrt{\sum_{i \in I} \mu_i Y_{ij}} + \Theta \sqrt{\sum_{i \in I} \sigma_i^2 Y_{ij}} \right\} + \sum_{i \in I} \lambda_i$$

$$\text{subject to} \quad Y_{ij} \le X_j \quad \forall i \in I, \forall j \in J$$

$$X_j \in \{0, 1\}, \quad \forall j \in J$$

$$Y_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J.$$

To solve this sub-problem, we can divide the objective function into three parts. The X part contains $\sum_{j \in J} f_j X_j$, the Y part contains $\sum_{i \in I} \{(d_{ij} - \lambda_i) Y_{ij} + K_j \sqrt{\sum_{i \in I} \mu_i Y_{ij}} + \Theta \sqrt{\sum_{i \in I} \sigma_i^2 Y_{ij}}\}$ and the λ part contains $\sum_{i \in I} \lambda_i$. Since the λ part has no decision variables, it can be ignored for the optimization. To minimize the objective function, we decide the value for X_j as follows: for each j, if f_j plus the minimum value of the Y part for that j is negative, we set X_j to 1 and set Y_{ij} to achieve the minimum value; else, we set X_j and Y_{ij} to 0. Therefore the problem reduces to finding the minimum value for the Y part. For given $j \in J$, let $b_i = d_{ij} - \lambda_i$, $c_i^1 = K_j^2 \mu_i$, $c_i^2 = \Theta^2 \sigma_i^2$ and $y_i = Y_{ij}$. Then the Y part minimization problem can be abstracted into

$$\text{minimize} \quad \sum_{i \in I} b_i y_i + \sqrt{\sum_{i \in I} c_i^1 y_i} + \sqrt{\sum_{i \in I} c_i^2 y_i}.$$

In [3] and [6], an efficient sorting method was introduced to solve this problem under the condition that $c_i^1 / c_i^2 =$ constant for all i. This allows the two square-root terms to be combined into one, which is requires for the algorithms in [3] and [6]. From the original notation, this condition means that the ratio of the demand variance to the demand mean is a constant for all retailers. This is approximately true for many realistic cases. However, in some cases demand variance can be influenced by factors other than the demand mean and that will violate the condition. One example is selling wine; the demand mean is mostly related to the population that drinks the wine while the demand variance can be influenced by the climate. People tend to buy more when the grapes are good because of good climate, so places with stable climate tend to have a lower demand variance.

As we mentioned in the last paragraph, the method in [3,6] works under the condition that $c_i^1 / c_i^2 =$ constant, let's say γ, for all i. Now the y part can be rewritten as follows:

$$\begin{aligned} &\sum_{i \in I} b_i y_i + \sqrt{\sum_{i \in I} c_i^1 y_i} + \sqrt{\sum_{i \in I} c_i^2 y_i} \\ &= \sum_{i \in I} b_i y_i + \sqrt{\sum_{i \in I} \gamma c_i^2 y_i} + \sqrt{\sum_{i \in I} c_i^2 y_i} \\ &= \sum_{i \in I} b_i y_i + (1 + \sqrt{\gamma}) \sqrt{\sum_{i \in I} c_i^2 y_i}. \end{aligned}$$

We get rid of one of the square root terms in this way. So the minimization problem of the Y terms can be written as:

$$(P) \quad \text{minimize} \quad \sum_{i \in I} b_i y_i + \sqrt{\sum_{i \in I} c_i y_i} \tag{5.1}$$

$$\text{subject to} \quad y_i \in \{0,1\} \quad \forall i \in I, \tag{5.2}$$

where $c_i \equiv c_i^2$

[3] and [6] proved that (P) can be solved efficiently by a sorting method, in $O(|I|\log|I|)$ time. The sorting method works as follows:

Let $I^- = \{i \in I | b_i < 0\}$ and $I_1^- = \{i \in I^- | b_i > 0\}$

1. Set $y_i = 0$ for all $i \in I^-$.

2. Set $y_i = 1$ for all $i \in I^-$ such that $c_i = 0$.

3. Sort the elements in I_1^- in increasing order of b_i/c_i.

4. For each $r \in \{0\} \cup I_1^-$,compute

$$S_r = \sum_{i=1}^{r} b_i + \sqrt{\sum_{i=1}^{r} c_i}.$$

(If $r = 0$, then $S_r = 0$.)

5. Choose the r that minimizes S_r and set $y_i = 1$ for $i = 1, ..., r$

The proof for this sorting method only uses the concave property of the square root function and the process also works if we replace the square root terms with any concave function terms.

So all the results hold if the functions are any concave function rather than square roots.

Without the condition that $c_i^1/c_i^2 =$ constant, we cannot combine the two square root terms into one in the Y part. So the sorting method provided by [3] and [6] does not work in these cases. Our focus is on identifying conditions under which the problem can be solved by the sorting method for problems with two or more square root terms.

5.2 Model Formulation

The general form of the multiple square root (MSR) minimization problem is as follows:

$$(PP)\text{minimize} \quad min \quad \sum_{i\in I} b_i y_i + \sqrt{\sum_{i\in I} c_i^1 y_i} + \sqrt{\sum_{i\in I} c_i^2 y_i} + ... + \sqrt{\sum_{i\in I} c_i^m y_i} \tag{5.3}$$

$$= \sum_{i\in I} b_i y_i + \sum_{t=1}^{m} \sqrt{\sum_{i\in I} c_i^t y_i}$$

$$\text{subject to} \quad y_i \in \{0, 1\}.$$

Throughout the analysis below, we assume that $c_i^t > 0$ for all $i \in I$ and $t \in \{1, ..., m\}$. In the case of the LMRP, this assumption holds as long as all of the demand variances are positive. If the assumption does not hold, one can approximate it by setting $c_i^t = \epsilon$, for small ϵ, if it equals 0. Furthermore, we will assume that $c_i^t \geq 1$ for all $i \in I$ and $t \in \{1, ..., m\}$. This assumption is not restrictive, since if $0 < c_i^t < 1$ we can multiply the objective function by some constant M without changing the optimal solution; c_i^t then becomes $M^2 c_i^t$. For large enough M, all coefficients are greater than or equal to 1.

Let $I^- = \{i \in I | b_i \leq 0\}$. Obviously, if $i \notin I^-$ then $y_i^* = 0$ in any optimal solution y^*. Assume without loss of generality that the elements of I^- are sorted such that

$$\frac{b_1}{\sum_{t=1}^{m} c_1^t} \leq \frac{b_2}{\sum_{t=1}^{m} c_2^t} \leq ... \leq \frac{b_n}{\sum_{t=1}^{m} c_n^t},$$

where $n = |I^-|$.

Consider the two square root problem, i.e., $m = 2$. As shown in [3] and [6], if $\frac{c_i^1}{c_i^2} = \text{constant}$ for all i, then the sorting method discussed above works. On the other hand, we know that without this constraint, the sorting method does not hold in general. The question we are interested in is whether the sorting method is guaranteed to work if the c_i^1/c_i^2 ratios are "close enough".

Suppose that $\frac{c_i^t}{c_i^1} = (1 + \alpha_i)c \quad t \in \{2, ..., m\}$ and $\forall i \in I$ with $\alpha_i \in [\underline{\alpha}, \bar{\alpha}]$ for some constants

α_i, $\underline{\alpha}$, $\bar{\alpha}$, c. Our aim is to identify values for $\underline{\alpha}$ and $\bar{\alpha}$, which guarantee that the sorting method works.

5.3 Model Analysis

In this section, we modify the proof of the correctness of the sorting method introduced by [3] and [6] to the MSR problem and discuss the bottleneck of the proof.

we assume without loss of generality that the elements of $I_1^- = \{i \in I^- | b_i < 0\}$ are indexed and sorted such that

$$\frac{b_1}{\sum_{t=1}^m c_1^t} \leq \frac{b_2}{\sum_{t=1}^m c_2^t} \leq ... \leq \frac{b_n}{\sum_{t=1}^m c_n^t}. \tag{5.4}$$

Conjecture 5.1. There exists an optimal solution y^* to (PP) such that the following property holds: if $y_k^* = 1$ for some $k \in I_1^-$, then $y_l^* = 1$ for all $l \in \{1, ..., k-1\}$.

Start of Proof: Let y^* be an optimal solution to (PP). Assume (for a contradiction) that $y_k^* = 1$ but $y_l^* = 0$ for some $l \in \{1, ..., k-1\}$. Assume that y^* is the optimal solution with the greatest number of components set to 1. We will find a new optimal solution with one more component set to one, thus contradicting our assumption.

Let $y_i' = \begin{cases} 1, & \text{if } i = l \\ y_i^*, & \text{otherwise,} \end{cases}$

$y_i'' = \begin{cases} 0, & \text{if } i = k \\ y_i^*, & \text{otherwise.} \end{cases}$

Let z^*; z'; z'' be the objective values corresponding to y^*; y'; y'', respectively. Let $R = \{i \in I^- \setminus \{k, l\} | y_i^* = 1\}$, and let $C^t = \sum_{i \in R} c_i^t$. Clearly,

$$z' - z^* = b_l + \sum_{t=1}^m \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t} \right] \tag{5.5}$$

$$z^* - z'' = b_k + \sum_{t=1}^m \left[\sqrt{C^t + c_k^t} - \sqrt{C^t} \right]. \tag{5.6}$$

Then from equation (5.5), we have:

$$\frac{z' - z^*}{\sum_{t=1}^m c_l^t} = \frac{b_t}{\sum_{t=1}^m c_l^t} + \frac{\sum_{t=1}^m (\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t})}{\sum_{t=1}^m c_l^t}. \tag{5.7}$$

Also, from equation (5.6), we have:

$$\frac{z^* - z''}{\sum_{t=1}^m c_k^t} = \frac{b_k}{\sum_{t=1}^m c_k^t} + \frac{\sum_{t=1}^m (\sqrt{C^t + c_k^t} - \sqrt{C^t})}{\sum_{t=1}^m c_k^t}. \tag{5.8}$$

Since $z' \leq z^*$ and $z'' \leq z^*$,

$$\frac{z' - z^*}{\sum_{t=1}^m c_l^t} > 0, \tag{5.9}$$

and

$$\frac{z^* - z''}{\sum_{t=1}^m c_k^t} < 0. \tag{5.10}$$

If we can identify a condition such that

$$\frac{z' - z^*}{\sum_{t=1}^m c_l^t} \leq \frac{z^* - z''}{\sum_{t=1}^m c_k^t}, \tag{5.11}$$

then we can say there exists a contradiction in inequalities (5.9), (5.10) and (5.11). Then Conjecture 5.1 would be proved.

From the convexity of the square root function, we have

$$\frac{\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t}}{c_l^t} \leq \frac{\sqrt{C^t + c_k^t} - \sqrt{C^t}}{c_k^t} \tag{5.12}$$

it is sufficient to find a condition that makes

$$\frac{\sum_{t=1}^m \left(\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t}\right)}{\sum_{t=1}^m c_l^t} \leq \frac{\sum_{t=1}^m \left(\sqrt{C^t + c_k^t} - \sqrt{C^t}\right)}{\sum_{t=1}^m c_k^t}. \tag{5.13}$$

An instance that satisfies inequality (5.13) must satisfy (5.11). Since $\frac{b_l}{\sum_{t=1}^m c_l^t} \leq \frac{b_k}{\sum_{t=1}^m c_k^t}$ by the way I^- is sorted.

Here comes the bottleneck, since inequality (5.13) doesn't hold for all cases. In Section 5.4 we give a sufficient condition that ensures inequality (5.13) holds, and in Section 5.5 we try an alternate approach.

5.4 A Sufficient Condition

As we assumed in Section 5.2, $\frac{c_i^t}{c_i^1} = (1+\alpha_i)c \quad t \in \{1, ..., m\}$ and $\forall i \in I$ with $\alpha_i \in [\underline{\alpha}, \bar{\alpha}]$. So we change the format of inequality (5.13) and associate it with our assumption.
First note that

$$\begin{aligned}
&\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t} \\
&= \frac{\left(\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t}\right)\left(\sqrt{C^t + c_k^t + c_l^t} + \sqrt{C^t + c_k^t}\right)}{\sqrt{C^t + c_k^t + c_l^t} + \sqrt{C^t + c_k^t}} \\
&= \frac{\left(C^t + c_k^t + c_l^t\right) - \left(C^t + c_k^t\right)}{\sqrt{C^t + c_k^t + c_l^t} + \sqrt{C^t + c_k^t}} \\
&= \frac{c_l^t}{\sqrt{C^t + c_k^t + c_l^t} + \sqrt{C^t + c_k^t}}. \qquad (5.14)
\end{aligned}$$

Given the assumption, we have:

$$c_l^t \leq c\,(1+\bar{\alpha})\,c_l^1$$

$$\sqrt{C^t + c_k^t + c_l^t} + \sqrt{C^t + c_k^t} \geq \sqrt{c}\sqrt{1-\underline{\alpha}}\left(\sqrt{C^1 + c_k^1 + c_l^1} + \sqrt{C^1 + c_k^1}\right).$$

Since all the terms are greater than zero, we have:

$$\frac{c_l^t}{\sqrt{C^t + c_k^t + c_l^t} + \sqrt{C^t + c_k^t}} \leq \frac{c\,(1+\bar{\alpha})\,c_l^1}{\sqrt{c}\sqrt{1-\underline{\alpha}}\left(\sqrt{C^1 + c_k^1 + c_l^1} + \sqrt{C^1 + c_k^1}\right)}. \qquad (5.15)$$

Simplifying the right hand side term, we have:

$$\begin{aligned}
&\frac{c\,(1+\bar{\alpha})\,c_l^1}{\sqrt{c}\sqrt{1-\underline{\alpha}}\left(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}\right)} \\
&=\sqrt{c}\frac{1+\bar{\alpha}}{\sqrt{1-\underline{\alpha}}}\frac{c_l^1\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right)}{\left(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}\right)\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right)} \\
&=\sqrt{c}\frac{1+\bar{\alpha}}{\sqrt{1-\underline{\alpha}}}\frac{c_l^1\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right)}{c_l^1} \\
&=\sqrt{c}\frac{1+\bar{\alpha}}{\sqrt{1-\underline{\alpha}}}\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right). \qquad (5.16)
\end{aligned}$$

Combining (5.14), (5.15) and (5.16), we have:

$$\sqrt{C^t+c_k^t+c_l^t}-\sqrt{C^t+c_k^t}\leq\sqrt{c}\frac{1+\bar{\alpha}}{\sqrt{1-\underline{\alpha}}}\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right).$$

This holds for any $t\in\{1,...,m\}$. Summing over t, we get:

$$\sum_{t=1}^{m}\left(\sqrt{C^t+c_k^t+c_l^t}-\sqrt{C^t+c_k^t}\right)\leq m\sqrt{c}\frac{1+\bar{\alpha}}{\sqrt{1-\underline{\alpha}}}\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right). \qquad (5.17)$$

Given the assumption, we also have:

$$\sum_{t=1}^{m}c_l^t\geq mc(1-\underline{\alpha})c_l^1, \qquad (5.18)$$

since all the terms are positive, inequality (5.17) and (5.18) imply

$$\begin{aligned}
&\frac{\sum_{t=1}^{m}\left(\sqrt{C^t+c_k^t+c_l^t}-\sqrt{C^t+c_k^t}\right)}{\sum_{t=1}^{m}c_l^t} \\
&\leq\frac{m\sqrt{c}\frac{1+\bar{\alpha}}{\sqrt{1-\underline{\alpha}}}\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right)}{mc(1-\underline{\alpha})c_l^1} \\
&=\frac{(1+\bar{\alpha})\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right)}{\sqrt{c}\sqrt{(1-\underline{\alpha})^3}c_l^1}. \qquad (5.19)
\end{aligned}$$

Similarly we have

$$
\begin{aligned}
&\sqrt{C^t + c_k^t} - \sqrt{C^t} \\
&= \frac{\left(\sqrt{C^t + c_k^t} - \sqrt{C^t}\right)\left(\sqrt{C^t + c_k^t} + \sqrt{C^t}\right)}{\sqrt{C^t + c_k^t} + \sqrt{C^t}} \\
&= \frac{(C^t + c_k^t) - (C^t)}{\sqrt{C^t + c_k^t} + \sqrt{C^t}} \\
&= \frac{c_k^t}{\sqrt{C^t + c_k^t} + \sqrt{C^t}}.
\end{aligned}
\tag{5.20}
$$

Given the assumption, we have:

$$
c_k^t \geq c\,(1 - \underline{\alpha})\, c_k^1
$$

$$
\sqrt{C^t + c_k^t} + \sqrt{C^t} \leq \sqrt{c}\sqrt{1 + \bar{\alpha}}\left(\sqrt{C^1 + c_k^1} + \sqrt{C^1}\right).
$$

Since all the terms are greater than zero, we have:

$$
\frac{c_k^t}{\sqrt{C^t + c_k^t} + \sqrt{C^t}} \geq \frac{c\,(1 - \underline{\alpha})\, c_k^1}{\sqrt{c}\sqrt{1 + \bar{\alpha}}\left(\sqrt{C^1 + c_k^1} + \sqrt{C^1}\right)}.
\tag{5.21}
$$

Simplifying the right hand side term, we have:

$$
\begin{aligned}
&\frac{c\,(1 - \underline{\alpha})\, c_k^1}{\sqrt{c}\sqrt{1 + \bar{\alpha}}\left(\sqrt{C^1 + c_k^1} + \sqrt{C^1}\right)} \\
&= \sqrt{c}\frac{1 - \underline{\alpha}}{\sqrt{1 + \bar{\alpha}}}\frac{c_k^1\left(\sqrt{C^1 + c_k^1} - \sqrt{C^1}\right)}{\left(\sqrt{C^1 + c_k^1} + \sqrt{C^1}\right)\left(\sqrt{C^1 + c_k^1} - \sqrt{C^1}\right)} \\
&= \sqrt{c}\frac{1 - \underline{\alpha}}{\sqrt{1 - \bar{\alpha}}}\frac{c_k^1\left(\sqrt{C^1 + c_k^1} - \sqrt{C^1}\right)}{c_k^1} \\
&= \sqrt{c}\frac{1 - \underline{\alpha}}{\sqrt{1 + \bar{\alpha}}}\left(\sqrt{C^1 + c_k^1} - \sqrt{C^1}\right).
\end{aligned}
\tag{5.22}
$$

Combining (5.20), (5.21) and (5.22), we have:

$$
\sqrt{C^t + c_k^t} - \sqrt{C^t} \geq \sqrt{c}\frac{1 - \underline{\alpha}}{\sqrt{1 + \bar{\alpha}}}\left(\sqrt{C^1 + c_k^1} - \sqrt{C^1}\right).
$$

This holds for any $t \in \{1, ..., m\}$. Summing over t, we get:

$$\sum_{t=1}^{m} \left(\sqrt{C^t + c_k^t} - \sqrt{C^t} \right) \geq m\sqrt{c}\frac{1-\underline{\alpha}}{\sqrt{1+\bar{\alpha}}} \left(\sqrt{C^1 + c_k^1} - \sqrt{C^1} \right). \tag{5.23}$$

Given the assumption, we also have:

$$\sum_{t=1}^{m} c_k^t \leq mc(1+\bar{\alpha})c_k^1, \tag{5.24}$$

since all the terms are positive, inequality (5.23) and (5.24) imply

$$\begin{aligned} &\frac{\sum_{t=1}^{m} \left(\sqrt{C^t + c_k^t} - \sqrt{C^t} \right)}{\sum_{t=1}^{m} c_k^t} \\ \geq\ & \frac{m\sqrt{c}\frac{1-\underline{\alpha}}{\sqrt{1+\bar{\alpha}}} \left(\sqrt{C^1 + c_k^1} - \sqrt{C^1} \right)}{mc(1+\bar{\alpha})c_k^1} \\ =\ & \frac{(1-\underline{\alpha}) \left(\sqrt{C^1 + c_k^1} - \sqrt{C^1} \right)}{\sqrt{c}\sqrt{(1+\bar{\alpha})^3}c_k^1}. \end{aligned} \tag{5.25}$$

From inequalities (5.19) and (5.25), we can see that if we have

$$\frac{(1+\bar{\alpha}) \left(\sqrt{C^1 + c_k^1 + c_l^1} - \sqrt{C^1 + c_k^1} \right)}{\sqrt{c}\sqrt{(1-\underline{\alpha})^3}c_l^1} \leq \frac{(1-\underline{\alpha}) \left(\sqrt{C^1 + c_k^1} - \sqrt{C^1} \right)}{\sqrt{c}\sqrt{(1+\bar{\alpha})^3}c_k^1} \tag{5.26}$$

then inequality (5.13) will hold. In other word, Conjecture (5.1) holds if we impose (5.16) as an additional condition.

Inequality (5.26) can be reformulated as

$$\frac{\sqrt{(1+\bar{\alpha})^5}}{\sqrt{(1-\underline{\alpha})^5}} \leq \frac{\frac{\sqrt{C^1+c_k^1}-\sqrt{C^1}}{c_k^1}}{\frac{\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}}{c_l^1}}$$

$$= \frac{\frac{\left(\sqrt{C^1+c_k^1}-\sqrt{C^1}\right)\left(\sqrt{C^1+c_k^1}+\sqrt{C^1}\right)}{c_k^1\left(\sqrt{C^1+c_k^1}+\sqrt{C^1}\right)}}{\frac{\left(\sqrt{C^1+c_k^1+c_l^1}-\sqrt{C^1+c_k^1}\right)\left(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}\right)}{c_l^1\left(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}\right)}}$$

$$= \frac{\frac{c_k^1}{c_k^1\left(\sqrt{C^1+c_k^1}+\sqrt{C^1}\right)}}{\frac{c_l^1}{c_l^1\left(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}\right)}}$$

$$= \frac{\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}}{\sqrt{C^1+c_k^1}+\sqrt{C^1}}. \tag{5.27}$$

For the right hand side of inequality (5.27), finding the minimum value of $\frac{\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}}{\sqrt{C^1+c_k^1}+\sqrt{C^1}}$ over all k, l, call it R^*, is easy to accomplish through a simple iterative procedure.

For every $t \in 1, ..., m$,

Step 1: Let $l = \underset{1 \leq i \leq k-1}{\operatorname{argmin}}\{c_i^t\}$.

Step 2: Let $k = \underset{2 \leq i \leq n}{\operatorname{argmin}}\{\frac{\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^1+c_k^1}}{\sqrt{C^1+c_k^1}+\sqrt{C^1}}\}$.

For the left hand side of inequality (5.17), the value is decided by $\frac{(1+\bar{\alpha})}{(1-\underline{\alpha})}$ and the smallest value of this fraction can also be determined from the dataset easily. Given our assumption in Section 5.2, the smallest value of $c\,(1+\bar{\alpha})$ should be the largest ratio of c_i^1 and c_i^t, for all $i \in I$ and $t \in 2, ..., m$; call it $\bar{r}$. Also, the largest value of $c\,(1+\bar{\alpha})$ should be the smallest ratio of $c_{i'}^1$ and $c_{i'}^{t'}$, for all $i' \in I$ and $t' \in 2, ..., m$; call it $\underline{r}$. So

$$\frac{(1+\bar{\alpha})}{(1-\underline{\alpha})} = \frac{c\,(1+\bar{\alpha})}{c\,(1-\underline{\alpha})} \geq \frac{\bar{r}}{\underline{r}}.$$

As a result, in a given dataset, if we have $\left(\frac{\bar{r}}{\underline{r}}\right)^{2.5} \leq R^*$, then inequality (5.27) holds for all k, l, and so inequality (5.13) holds. In this case conjecture (5.1) holds, and the sorting method method can be applied to the data set.

The following theorem summaries our analysis:

Theorem 5.4.1. *If a data set has the property that* $\left(\frac{\bar{r}}{\underline{r}}\right)^{2.5} \leq R^*$, *then there exists an optimal solution* y^* *to (PP) such that the following property holds: if* $y_k^* = 1$ *for some* $k \in I_1^-$, *then* $y_l^* = 1$ *for all* $l \in \{1, ..., k-1\}$.

If the condition in Theorem (5.4.1) holds, then the sorting method can be applied.

5.5 An Alternate Analytical Approach

There is another direction for finding the condition. Continuing the logic from the Section 5.3, let's start from the simplest case, the two square root case.

First we subtract the left hand side from the right hand side of inequality (5.13);

$$\frac{(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^2+c_k^2+c_l^2}-\sqrt{C^1+c_k^1}-\sqrt{C^2+c_k^2})}{c_l^1+c_l^2} - \frac{(\sqrt{C^1+c_k^1}+\sqrt{C^2+c_k^2}-\sqrt{C^1}-\sqrt{C^2})}{c_k^1+c_k^2}$$
$$= \frac{(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^2+c_k^2+c_l^2}-\sqrt{C^1+c_k^1}-\sqrt{C^2+c_k^2})(c_k^1+c_k^2)-(\sqrt{C^1+c_k^1}+\sqrt{C^2+c_k^2}-\sqrt{C^1}-\sqrt{C^2})(c_l^1+c_l^2)}{(c_l^1+c_l^2)(c_k^1+c_k^2)}.$$

The denominator is larger than zero as c_k and c_l are larger than zero. For the numerator, we need to compare $(\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^2+c_k^2+c_l^2})(c_k^1+c_k^2)+(\sqrt{C^1}+\sqrt{C^2})(c_l^1+c_l^2)$ and $(\sqrt{C^1+c_k^1}+\sqrt{C^2+c_k^2})(c_k^1+c_k^2+c_l^1+c_l^2)$.

From here we can see that if we do not add a condition on the relation between $c_k^1+c_k^2$ and $c_l^1+c_l^2$, the difference between the previous terms can be either positive or negative. So we first add the simple condition that $\sum_{t=1}^m c_l^t = \sum_{t=1}^m c_k^t$.

So the problem changes to comparing $\sqrt{C^1+c_k^1+c_l^1}+\sqrt{C^2+c_k^2+c_l^2}+\sqrt{C^1}+\sqrt{C^2}$ and $2(\sqrt{C^1+c_k^1}+\sqrt{C^2+c_k^2})$. As these terms are all positive, we can square them without changing their magnitude relation.

The left hand side becomes

$$C^1+c_k^1+c_l^1+C^2+c_k^2+c_l^2+C^1+C^2+2\Big[(\sqrt{C^1+c_k^1+c_l^1}\sqrt{C^2+c_k^2+c_l^2})+(\sqrt{C^1}\sqrt{C^2})+$$
$$(\sqrt{C^1+c_k^1+c_l^1}\sqrt{C^1})+(\sqrt{C^2+c_k^2+c_l^2}\sqrt{C^2})+(\sqrt{C^1+c_k^1+c_l^1}\sqrt{C^2})+(\sqrt{C^2+c_k^2+c_l^2}\sqrt{C^1}).$$

The right hand side becomes

$4(C^1 + c_k^1 + C^2 + c_k^2 + 2(\sqrt{C^1 + c_k^1}\sqrt{C^2 + c_k^2})),$

which can be written as the same pattern as the left hand side,

$C^1 + c_k^1 + C^2 + c_k^2 + C^1 + c_k^1 + C^2 + c_k^2 + 2\Big[(\sqrt{C^1 + c_k^1}\sqrt{C^2 + c_k^2}) + (\sqrt{C^1 + c_k^1}\sqrt{C^2 + c_k^2}) +$
$(\sqrt{C^1 + c_k^1}\sqrt{C^1 + c_k^1}) + (\sqrt{C^2 + c_k^2}\sqrt{C^2 + c_k^2}) + (\sqrt{C^1 + c_k^1}\sqrt{C^2 + c_k^2}) + (\sqrt{C^1 + c_k^1}\sqrt{C^2 + c_k^2})$

The difference between these two sides is

$(\sqrt{C^1 + c_k^1 + c_l^1}\sqrt{C^2 + c_k^2 + c_l^2}) + (\sqrt{C^1}\sqrt{C^2}) + (\sqrt{C^1 + c_k^1 + c_l^1}\sqrt{C^1}) + (\sqrt{C^2 + c_k^2 + c_l^2}\sqrt{C^2}) +$
$(\sqrt{C^1 + c_k^1 + c_l^1}\sqrt{C^2}) + (\sqrt{C^2 + c_k^2 + c_l^2}\sqrt{C^1}) - 4(\sqrt{C^1 + c_k^1}\sqrt{C^2 + c_k^2}).$

This term's value is near 0 but can be positive or negative for different data sets. We tried to add some conditions to make this term always negative, but no sufficient condition was found. However, if we only impose the simple condition that $\sum_{t=1}^{m} c_l^t = \sum_{t=1}^{m} c_k^t$, the sorting method often finds the optimal solution, and when it does not, the error is usually small. This will be shown in the next section.

5.6 A Lower Bound Based on the Sufficient Condition

As we discussed in Theorem 5.4.1, if a data set has the property that $\left(\frac{\bar{r}}{\underline{r}}\right)^{2.5} \leq R^*$, then there exists an optimal solution y^* to (PP) such that the following property holds: if $y_k^* = 1$ for some $k \in I_1^-$, then $y_l^* = 1$ for all $l \in \{1, ..., k-1\}$. From the Theorem we can see that in a data set, the value of $R^* \left(\frac{\bar{r}}{\underline{r}}\right)^{-2.5}$ will decide whether this data set holds the property. Let's name this number as R. So when $R \geq 1$, the property holds and the sorting method can be applied to this data set. In this section, we discuss a method to get a lower bound through the sorting method when $0 < R < 1$ and the gap between the lower bound and optimal solution.

Conjecture 5.2. There exists a lower bound $z_{lb} = z' + \left(\frac{1-R}{R}\right) min\{b_i\,(i \in 1, ...k-1)\}$ to (PP) such that z' is the objective value of a feasible solution y' that the following property holds: if $y_k^{\prime=1}$ for some $k \in I_1^-$, then $y^{\prime l=1}$ for all $l \in \{1, ..., k-1\}$.

Start of Proof: Let y^* be an optimal solution to (PP) and z^* is it's objective value. Assume (for a contradiction) that $y_k^* = 1$ but $y_l^* = 0$ for some $l \in \{1, ..., k-1\}$ and $z^* < z_{lb}$. We will find a feasible solution y' with the following property: if $y_k^{'=1}$ for some $k \in I_1^-$, then $y^{'=1}_l$ for all $l \in \{1, ..., k-1\}$ that breaks the inequality, thus contradicting our assumption.

From Equation(5.5) we have:

$$z' - z^* = b_l + \sum_{t=1}^{m} \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t} \right]$$

,

as $0 < R < 1$ we have

$$\begin{aligned}
R\left(z' - z^*\right) &= R\left(b_l + \sum_{t=1}^{m} \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t} \right]\right) \\
Rz' - Rz^* &= Rb_l + R\sum_{t=1}^{m} \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t} \right] \\
Rz' - Rz^* + (1-R)\, b_l &= b_l + R\sum_{t=1}^{m} \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t} \right]. \qquad (5.28)
\end{aligned}$$

From the Assumption,

$$\begin{aligned}
& z^* < z_{lb} \\
& Rz^* < Rz' + (1-R)\, min\{b_i\, (i \in 1, ...k-1)\} \\
& Rz^* < Rz' + (1-R)\, b_l \\
& Rz' - Rz^* + (1-R)\, b_l > 0. \qquad (5.29)
\end{aligned}$$

From Equation(5.28) we have

$$\frac{Rz' - Rz^* + (1-R)\, b_l}{\sum_{t=1}^m c_l^t} = \frac{b_l}{\sum_{t=1}^m c_l^t} + R\frac{\sum_{t=1}^m \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t} \right]}{\sum_{t=1}^m c_l^t}. \qquad (5.30)$$

Also, we have Equation (5.8)

$$\frac{z^* - z''}{\sum_{t=1}^m c_k^t} = \frac{b_k}{\sum_{t=1}^m c_k^t} + \frac{\sum_{t=1}^m (\sqrt{C^t + c_k^t} - \sqrt{C^t})}{\sum_{t=1}^m c_k^t}. \qquad (5.31)$$

Based on the way we sort the data,

$$\frac{b_l}{\sum_{t=1}^m c_l^t} \leq \frac{b_k}{\sum_{t=1}^m c_k^t}. \tag{5.32}$$

According to Inequalities (5.19), (5.25), and (5.26), we can imply that,

$$R\frac{\sum_{t=1}^m \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t}\right]}{\sum_{t=1}^m c_l^t} \leq \frac{\sum_{t=1}^m (\sqrt{C^t + c_k^t} - \sqrt{C^t})}{\sum_{t=1}^m c_k^t}. \tag{5.33}$$

Combining Inequalities (5.32) and (5.33) we have,

$$\frac{b_l}{\sum_{t=1}^m c_l^t} + R\frac{\sum_{t=1}^m \left[\sqrt{C^t + c_k^t + c_l^t} - \sqrt{C^t + c_k^t}\right]}{\sum_{t=1}^m c_l^t} \leq \frac{b_k}{\sum_{t=1}^m c_k^t} + \frac{\sum_{t=1}^m (\sqrt{C^t + c_k^t} - \sqrt{C^t})}{\sum_{t=1}^m c_k^t},$$

so

$$\frac{Rz' - Rz^* + (1-R)\, b_l}{\sum_{t=1}^m c_l^t} \leq \frac{z^* - z''}{\sum_{t=1}^m c_k^t}. \tag{5.34}$$

From Inequality (5.29),

$$\frac{Rz' - Rz^* + (1-R)\, b_l}{\sum_{t=1}^m c_l^t} > 0. \tag{5.35}$$

Clearly,

$$\frac{z^* - z''}{\sum_{t=1}^m c_k^t} \leq 0. \tag{5.36}$$

There exists a contradiction in inequalities (5.34), (5.35) and (5.36). As a result, Conjecture 5.2 is proved.

5.7 Test on Random Samples

We generate the parameters in our random samples using uniformly distributed random numbers. b_i was generated from $U[-10, 0]$ and all c_i^k were generated from $U[0, 1]$ and we choose $|I| = 10$ and $|m| = 2, 3, 4$. There are 600 instances in total. We get the optimal value of the sample instances by enumerating each of the feasible solutions and choosing the ones with the smallest objective value. Then we apply the sorting method to the same instance and compare the result with the enumeration method. Both methods were implemented in MATLAB.

We can see that in most of the two square root cases, 94 out of 100, the sorting method find the optimal solution. In these samples, 42 instances satisfy the sufficient condition (5.17) and for each of these instances, the sorting method finds the optimal solution, as predicted by Theorem (5.4.1). In the three square root cases, the sorting method works in 73 out of 100 data sets and 15 of them meet condition (5.17). When there are four square root terms, the sorting method works in 56 out of 100 data sets, and there are only 6 cases that meet condition (5.17). The results are shown in Table 5.2. This shows that the sufficient condition is correct, but not many data sets satisfy it. However, this condition allows slight changes in the ratio of the multiple-square-root parameters, which is typical of realistic data sets. For further research, relaxing the condition so that it applies to more realistic data sets can be a direction.

For the alternate analytical approach in Section 5.5; the accuracy rates are shown in Table 5.1. We can see that as the complexity of the problem goes up with more square roots, the accuracy rate goes down. However, the rates are acceptable when the number of square roots is not large. For future research, we will search for additional sufficient conditions to guarantee the optimality of the sorting method.

Number of square root terms	Accuracy rate	Average error for all instances	Average error for failed instances
2	0.93	0.9875	0.8514
3	0.87	0.9493	0.7483
4	0.62	0.8853	0.7048

Table 5.1: Accuracy rate for the alternate analytical approach

Number of square root terms	Satisfaction rate	Average error for all instances	Average error for failed instances
2	0.42	0.9856	0.7314
3	0.15	0.9347	0.6026
4	0.06	0.8653	0.6892

Table 5.2: Satisfaction rate for the sufficient condition

Chapter 6

Conclusions and Future Research

In this thesis, we discuss two problems that involve concave minimization problems with binary variables.

Our linearization of the LMRP requires longer solution time on average than the Lagrangian method does. However, it performs better in some special instances.

As for the MSR problem, we discuss a sufficient condition that guarantees the sorting method works, and an alternate analytical approach that makes the sorting method often find the optimal solution while the average error is low.

For future research, we would like to determine under what conditions the linearization method will have a shorter solution time than the Lagrangian method does. Searching for relatively relaxed sufficient conditions to guarantee that the sorting method works for the MSR problem is also a direction.

Bibliography

[1] Mark S. Daskin. Application of an expected covering model to emergency medical service system design. *Decision Sciences*, 13(3):416–439, 1982.

[2] Mark S. Daskin. A maximum expected covering location model: formulation, properties and heuristic solution. *Transportation Science*, 17(1):48–70, 1983.

[3] Mark S. Daskin, Collette R. Coullard, and Zuo-Jun Max Shen. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, 110(1-4):83–106, 2002.

[4] Leyla Ozsen, Collette R. Coullard, and Mark S. Daskin. Capacitated warehouse location model with risk pooling. *Naval Research Logistics*, 55(4):295–312, 2008.

[5] Zuo-Jun Max Shen. Integrated supply chain design models: a survey and future research directions. *Journal of Industrial and Management Optimization*, 3(1):1, 2007.

[6] Zuo-Jun Max Shen, Collette R. Coullard, and Mark S. Daskin. A joint location-inventory model. *Transportation Science*, 37(1):40–55, 2003.

[7] Jia Shu, Chung-Piaw Teo, and Zuo-Jun Max Shen. Stochastic transportation-inventory network design problem. *Operations Research*, 53(1):48–60, 2005.

[8] Lawrence V. Snyder, Mark S. Daskin, and Chung-Piaw Teo. The stochastic location model with risk pooling. *European Journal of Operational Research*, 179(3):1221–1238, 2007.

Vita

The author was born in China in 1991. He was awarded the bachelor of science degree in Electronic Engineering in Peking University in 2013. Following that, he spent two and a half years studying in Industrial System Engineering department in Lehigh University. By 2017, he finished his M.S. degree in Lehigh University.