BEIJING NORMAL UNIVERSITY - HONG KONG BAPTIST UNIVERSITY UNITED INTERNATIONAL COLLEGE

GROUP PROJECT

The Optimization for the pipe purchasing and shipping plan

Author:
Henry CHEN , Charles LIN ,
Rain SHI , Andy WANG ,
Hubert LI , Tony YE

Supervisor:
Dr. DENG

A thesis submitted in fulfillment of the requirements for the Optimization Project of sophomore

in the

Nonlinear Oilflower Group 2
Department of Science and Technology - Statistic

January 3, 2022

BEIJING NORMAL UNIVERSITY - HONG KONG BAPTIST UNIVERSITY
UNITED INTERNATIONAL COLLEGE

Abstract

Faculty Name
Department of Science and Technology - Statistic

sophomore

The Optimization for the pipe purchasing and shipping plan

by Henry CHEN , Charles LIN , Rain SHI , Andy WANG , Hubert LI , Tony YE

This project is related to the programming of minimize the total cost of the transportation and lay the nature gas pipelines. Pipelines should be linked from A_1 to A_{15} as shown in Figure 3.1. The steel factories from S_1 to S_r are able to satisfy the requirement of pipes as the price shown in the table 3.1. The pipes from each factories should be at least 500 km.Assuming that the pipeline transported to the location point A_j can only be used to lay the interval between A_{j-1} and A_{j+1}, and the interval must include A_j, that is, such pipelines can be laid to A_{j-1}, but cannot exceed A_{j-1}, or Lay similarly to A_{j+1}, but cannot exceed A_{j+1}, or both directions. For such pipeline movement, the transportation cost per unit pipeline is 0.1/km, and the movement distance is less than 1km. Calculated by 1km, that is, 0.1 km per unit. For the sake of simplicity, we further assume that no pipes will be wasted during the construction process, so the total amount of pipes required is equal to the length of the entire pipe $A_1 \to A_2 \to ... \to A_{15}$. In addition, all required pipes have the same size.

In this project, we need to consider the cost of the pipeline, the transportation fee and the laying cost. For the factory, we need to determine whether the factory will produce the pipeline or not. For the transportation procession, we need the consider the direction along the traffic lines, including the railway and the high-way. And for the laying cost, we need to consider the laying direction and the corresponding length.

Contents

Abstract iii

1 Modeling Programming 1
- 1.1 Several model assumption . . . 1
- 1.2 Mathematical model . . . 1
  - 1.2.1 Variable name and corresponding meaning . . . 1
  - 1.2.2 Model . . . 2
  - 1.2.3 Model interpretation . . . 4
- 1.3 Optimization Method for Model . . . 4
  - 1.3.1 Optimization processing . . . 5
- 1.4 Solution for the question . . . 5
  - 1.4.1 The optimal solution for mathematical model . . . 5
- 1.5 Result analysis . . . 5
  - 1.5.1 Plausibility test . . . 5

A Appendix 7
- A.1 Appendix a . . . 7
  - A.1.1 Optimization Theorems . . . 7
  - A.1.2 References . . . 7
- A.2 Appendix b . . . 8
  - A.2.1 Figures . . . 8
  - A.2.2 Codes . . . 9
  - A.2.3 Tables . . . 12

Chapter 1

Modeling Programming

1.1 Several model assumption

As the Abstract included, the question is too complicated if we consider the question like the real life, so for modeling a suitable and enough simple model, we made these assumptions:

(1) The traffic line is bidirectional for all the transportations .

(2) After arriving at the destination, it is transported and installed section by section every 1km.Insufficiency is calculated as 1km.

(3) Every factories we used must produce at least 500km pipes.

(4) During the construction process, no pipes shall be wasted and hence the total amount of pipes required is equal to the length of the entire pipeline $A_1 \to A_2 \to ... \to A_{15}$.

(5) The pipes made by factories are totally transported and used.

1.2 Mathematical model

For this project, it is related to the production,transportation, and installation cost. For production part, there are seven factories can produce the pipelines. For different factories, they have different capacity upper bound and the sell price per unit in the given table. For the transportation part, we can choose different pathes contains the length of highway and railway we use. These two transportation methods are have different cost per unit, the highway is 0.1 unit money per kilometer and the railway is 0.063 unit money per kilometer (one unit money is 10000 yuan).And after transporting the pipelines to the installation destination, we need to decide the length of pipelines in two directation we will install.

1.2.1 Variable name and corresponding meaning

B_i $\sim$ The amount of the pipeline produced by S_i factory.

C_j $\sim$ The cost of transportation 1 km in this path.

K_i $\sim$ The capacity for factory S_i

P_i $\sim$ The price of pipelines produced by factory S_i

$Q_{a,b,j}$ $\sim$ The amount of the pipelines need to be transported from point a to b, in path C_j.

$W_{A_k \leftrightarrow A_{k+1}}$ $\sim$ The amount of money we need use in laying out the pipeline. And the direction may be left or right.

$d_{A_i \to A_k}$ $\sim$ The length of pipeline will lay from the node A_i to $A_k . k = i-1, i+1$

1.2.2 Model

Objective functions:

Minimize: $f_1 + f_2 + f_3$

production cost + transportation cost+ Laying cost

$$f_1 = \sum_{i=1}^{7} B_i P_i$$

$$f_2 = \sum_j Q_{a,b,j} C_j$$

$$f_3 = \sum_k W_{A_k \leftrightarrow A_{k+1}}$$

$$W_{A_k \leftrightarrow A_{k+1}} = \begin{cases} 0.1 & d_{A_j \leftrightarrow A_{j+1}} \le 1 \\ 0.1 \times \frac{[d_{A_j \leftrightarrow A_{j+1}}] \times \{1 + [d_{A_j \leftrightarrow A_{j+1}}]\}}{2} + 0.1 & d_{A_j \leftrightarrow A_{j+1}} > 1 \end{cases}$$

Subject to:

1. The amount used pipelines must be $\le$ than the amount of bought pipelines.

$$\sum_{i=1}^{7} B_i \ge \sum_{j=1}^{14} ||A_j A_{j+1}|| = 5171$$

2. For any node, the amount of transporting pipelines are holds.

N_1: $Q_{N_1 A_1} + Q_{N_1 A_2} = Q_{N_2 A_1}$
N_2: $Q_{N_2 A_1} + Q_{N_2 A_3} = Q_{N_3 N_2}$
N_3: $Q_{N_3 N_2} + Q_{N_3 A_4} = Q_{N_7 N_3}$
N_4: $Q_{N_4 A_5} + Q_{N_4 A_6} = Q_{N_5 N_4}$
N_5: $Q_{N_6 N_5} = Q_{N_5 N_4} + Q_{N_5 A_7}$
N_6: $Q_{N_6 N_5} + Q_{N_6 A_7} = Q_{N_7 N_6} + B_1 - Q_{N_6 N_7}$
N_7: $Q_{N_7 N_3} + Q_{N_7 N_6} + Q_{N_7 N_8} + Q_{N_7 A_8} = B_2 + Q_{N_6 N_7} + Q_{N_8 N_7}$
N_8: $Q_{N_8 N_7} + Q_{N_8 A_9} + Q_{N_8 N_9} = B_3 + Q_{N_7 N_8} + Q_{N_9 N_8}$
N_9: $Q_{N_9 N_8} + Q_{N_9 A_{10}} + Q_{N_9 N_{10}} = Q_{N_{10} N_9} + Q_{N_8 N_9}$
N_{10}: $Q_{N_{10} N_9} + Q_{N_{10} N_{12}} + Q_{N_{10} N_{11}} = B_4 + Q_{N_9 N_{10}} + Q_{N_{11} N_{10}} + Q_{N_{12} N_{10}}$
N_{11}: $Q_{N_{11} A_{11}} + Q_{N_{11} N_{10}} = B_5 + Q_{N_{10} N_{11}}$
N_{12}: $Q_{N_{12} N_{10}} + Q_{N_{12} A_{12}} + Q_{N_{12} N_{13}} = Q_{N_{13} N_{12}} + Q_{N_{10} N_{12}}$
N_{13}: $Q_{N_{13} N_{12}} + Q_{N_{13} A_{13}} + Q_{N_{13} N_{14}} = Q_{N_{12} N_{13}} + Q_{N_{14} N_{13}}$

N_{14}: $Q_{N_{14}N_{13}} + Q_{N_{14}N_{16}} + Q_{N_{14}N_{15}} + Q_{N_{14}A_{14}} = Q_{N_{13}N_{14}} + Q_{N_{16}N_{14}} + Q_{N_{15}N_{14}}$
N_{15}: $Q_{N_{15}A_{14}} + Q_{N_{15}N_{14}} = B_6 + Q_{N_{14}N_{15}}$
N_{16}: $Q_{N_{16}N_{14}} + Q_{N_{16}A_{15}} + Q_{N_{16}N_{17}} = Q_{N_{14}N_{16}} + Q_{N_{17}N_{16}}$
N_{17}: $Q_{N_{17}A_{15}} + Q_{N_{17}N_{16}} = Q_{N_{16}N_{17}} + B_7$

3. the amount of the production should satisfy the constrains from capacity.

$$B_i = \begin{cases} \geq 500 & \text{if we choose } S_i \\ 0 & \text{if we don't choose } S_i \end{cases}$$

$$B_i \leq K_i, \quad i = 1,2,3,4,5,6,7$$

If we don't choose S_1 : $B_1 = Q_{N_6N_7} = 0$
If we don't choose S_2 : $B_2 = 0$
If we don't choose S_3 : $B_3 = 0$
If we don't choose S_4 : $B_4 = 0$
If we don't choose S_5 : $B_5 = 0$
If we don't choose S_6 : $B_6 = Q_{N_{15}N_{14}} = 0$
If we don't choose S_7: $B_7 = Q_{N_{17}N16} = 0$

4. Laying the pipelines:

4.1 Garantee that the pipline is properly connected in any node $A_j, \quad j = 1, ..., 15.$

$d_{A_1\to A_2} + d_{A_2\to A_1} = 104$
$d_{A_2\to A_3} + d_{A_3\to A_2} = 301$
$d_{A_3\to A_4} + d_{A_4\to A_3} = 750$
$d_{A_4\to A_5} + d_{A_5\to A_4} = 606$
$d_{A_5\to A_6} + d_{A_6\to A_5} = 194$
$d_{A_6\to A_7} + d_{A_7\to A_6} = 205$
$d_{A_7\to A_8} + d_{A_8\to A_7} = 201$
$d_{A_8\to A_9} + d_{A_9\to A_8} = 680$
$d_{A_{10}\to A_9} + d_{A_9\to A_{10}} = 480$
$d_{A_{10}\to A_{11}} + d_{A_{11}\to A_{10}} = 300$
$d_{A_{12}\to A_{11}} + d_{A_{11}\to A_{12}} = 220$
$d_{A_{12}\to A_{13}} + d_{A_{13}\to A_{12}} = 210$
$d_{A_{13}\to A_{14}} + d_{A_{14}\to A_{13}} = 420$
$d_{A_{15}\to A_{14}} + d_{A_{14}\to A_{15}} = 500$

4.2 The relationship between the amount of pipelines having been transportedd to the node $A_j, \quad j = 1, ..., 15.$

$d_{A_1\to A_2} = Q_{N_1A_1}$
$d_{A_2\to A_1} + d_{A_2\to A_3} = Q_{N_1A_2}$
$d_{A_3\to A_2} + d_{A_3\to A_4} = Q_{N_2A_3}$
$d_{A_4\to A_3} + d_{A_4\to A_5} = Q_{N_3A_4}$
$d_{A_5\to A_4} + d_{A_5\to A_6} = Q_{N_4A_5}$
$d_{A_6\to A_5} + d_{A_6\to A_7} = Q_{N_4A_6}$
$d_{A_7\to A_6} + d_{A_7\to A_8} = Q_{N_5A_7} + Q_{N_6A_7}$
$d_{A_8\to A_7} + d_{A_8\to A_9} = Q_{N_7A_8}$
$d_{A_9\to A_8} + d_{A_9\to A_{10}} = Q_{N_8A_9}$
$d_{A_{10}\to A_9} + d_{A_{10}\to A_{11}} = Q_{N_9A_{10}}$
$d_{A_{11}\to A_{10}} + d_{A_{11}\to A_{12}} = Q_{N_{11}A_{11}}$
$d_{A_{12}\to A_{11}} + d_{A_{12}\to A_{13}} = Q_{N_{12}A_{12}}$

$d_{A_{13}\to A_{12}} + d_{A_{13}\to A_{14}} = Q_{N_{13}A_{13}}$
$d_{A_{14}\to A_{13}} + d_{A_{14}\to A_{15}} = Q_{N_{15}A_{14}} + Q_{N_{14}A_{14}}$
$d_{A_{15}\to A_{14}} = Q_{N_{17}A_{15}}$

To simplify the expression, we use daaij to substitude $d_{A_i\to A_j}$ in code. To simplify the expression, we use Qnaij to substitude $Q_{N_iA_j}$ in code.

1.2.3 Model interpretation

We can find that for every factory, if we choose them to produce the pipelines, the least production amount should be 500. If we don't choose the factory, its production will be 0. This means there must be a binary variable in this part.

Next, we will give a programming to the transportation. For the traffic, we have two options: by train or by lorry. And for every traffic line, we consider it to be bidirectional. That is, a line has two terminal points A and B. The items can be transported from B to A or A to B. We classify two different types of node: unidirectional node and bidirectional node. For unidirectional node, it only have one enter path and several exit paths. For bidirectional node, it contains some path that can be used to take the pipelines to the node and take it out of the node. In brief, the paths only connected with unidirectional road have the unique transportation direction. And the paths connected to the bidirectional path may have two directions transportation. The most obvious difference between bidirectional and unidirectional node is that for unidirectional node, there is no factories among all the paths and the destination.

After this, we have moved the pipeline to the destination, and we need to install it. We will make an assumption that we need to install the pipelines one unit (1 km) in one time. So the total installation fee is like a form of arithmetic sequence. And we may also use the round down function to get the cost of the installation.

1.3 Optimization Method for Model

Interior Point Method:

For the optimization, to begin with, we choose a point in the feasible region as we defined in the model, and then used Interior Point Method to solve it. Unfortunately, the solutions shown by the Matlab are not feasible even if we get tens of thousands of iterations. Last, trough searching on the internet, we find that there are some bugs if we use the fmincon solver, and $InteriorPointMethod$ also cannot be used in equal situation. It means that this method does not suit the model we made.

Lagrange Multiplier Method:

As an optimization algorithm, Lagrangian multiplier method is mainly used to solve constrained optimization problems. Its basic idea is to transform a constrained optimization problem containing variables and constraints as we shown in the last page by introducing Lagrangian multipliers. It is an unconstrained optimization problem with variables. The Lagrangian multiplier method starts from the mathematical meaning and establishes extreme conditions by introducing Lagrangian multipliers.

1.3.1 Optimization processing

We are trying to use MATLAB to solve this problem. But for MATLAB, the constrains are too many to solve. We use the 'fmincon' solver to get the solution. But when we observe the final answer, we found that there are some negative part in it. It made us feel shocked, because the negative part make the solution infeasible. We cannot find any error in it, so we use lingo to get the final answer.(The matlab code will also be packaged in the folder, but we may not list them in the report.)

See Appendix a A.1.1 and b A.2.2(Codes for LINGO) for details.

1.4 Solution for the question

Objective value: 1305033.

Objective bound: 1305033.

(Dimension: Units money)

1.4.1 The optimal solution for mathematical model

As the tables shown in the Appendix b A.2.3

1.5 Result analysis

In the optimal solution we made, the infeasibilities we have is $0.1136868e^{-12}$, which means that the solution is very Convincing. Also as the Fig. 3.2 shows for the extended solver steps and total solver iterations, the numbers are very normal in this question. We can accept these outcomes, and make a conclusion that the optimal solution is 1305033.

1.5.1 Plausibility test

We have formulated 5 hypotheses as far as possible while preserving the reality to facilitate our model building. Of course, through the final results and the above-mentioned evaluation of the credibility of the results, we combined with real-life cases and agreed that when the pipeline length is about 5000km, the final amount of 13050330000 is in line with the actual situation.

Appendix A

Appendix

A.1 Appendix a

A.1.1 Optimization Theorems

Lagrange Multiplier Method(KKT):

Among the problems with constraints, we are more inclined to transform them into unconstrained problems. In mathematical optimization problems, the Lagrangian multiplier method is a method of finding the extreme value of a multivariate function whose variables are restricted by one or more conditions. This method transforms an optimization problem with n variables and k constraints into an extreme value problem with n + k variables, the variables of which are not subject to any constraints. This method introduces a new scalar unknown, namely the Lagrangian multiplier (the coefficient of each vector in the linear combination of the gradient of the constraint equation).

KKT condition: is necessary and sufficient condition for a nonlinear programming (Nonlinear Programming) problem to have an optimal solution under certain regular conditions. This is the result of a generalized Lagrangian multiplier.

A.1.2 References

Lagrange Multiplier

MATLAB-fmincon Function Explaining

Interior Point Method

A.2 Appendix b

A.2.1 Figures

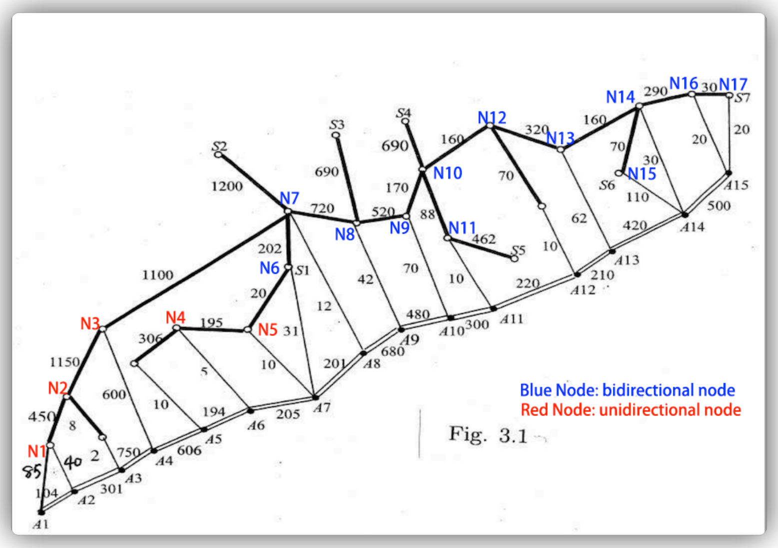

Table 3.1: Production Capacity and Sell Price

Factory S_i	S1	S2	S3	S4	S5	S6	S7
Capacity K_i	800	800	1000	2000	2000	2000	3000
Price P_i	160	155	155	160	155	150	160

FIGURE A.1: where K_i is the maximum production capacity (in km of length), and P_i is the price for 1 km (i.e. 1 unit) of pipes(the money unit used is 10,000 Yuan) for the product of factory S_i.

FIGURE A.2: The optimal solution for the model

A.2.2 Codes

```
min=160 * B1 + 155 * B2 + 155 * B3 + 160 * B4 + 155 * B5 + 150 * B6
+ 160 * B7+Qna11 * 85 * 0.065 + Qna12 * 40 * 0.1
+Qna21* 450* 0.065 + Qna23 * (8 * 0.065 + 2 * 0.1)+Qnn32 * 1150* 0.065
+ Qna34 * 600 * 0.1+Qna45 *(306 * 0.065 + 10 * 0.1) + Qna46 * 5 * 0.1
+Qnn54 * 195* 0.065 + Qna57 * 10 * 0.1+Qnn65 * 20 * 0.065
+ Qna67 * 31 * 0.1 + (Qnn76 + Qnn67) * 202 * 0.065
+Qnn73 * 1100* 0.065 + Qna78 * 12 * 0.1
+ (Qnn78 + Qnn87) * 720 * 0.065+Qna89 * 42 * 0.1
 + (Qnn89 + Qnn98) * 520 * 0.065+Qna910 * 70 * 0.1
+(Qnn910 + Qnn109) * 170 * 0.065+Qna1111* 10 * 0.1
 + (Qnn1110 + Qnn1011) * 88 *0.065+Qna1212* (70 * 0.065 + 10 * 0.1)
+(Qnn1210 + Qnn1012) * 160 * 0.065 + (Qnn1213 + Qnn1312) * 320 * 0.065
+Qna1313* 62 * 0.1 + (Qnn1314 + Qnn1413) * 160 * 0.065
+Qna1414* 30 * 0.1 + (Qnn1416 + Qnn1614) * 290 * 0.065
+ (Qnn1415 + Qnn1514) * 70 * 0.065
+Qna1514* 110 * 0.1+Qna1615* 20 * 0.1
+ (Qnn1617 + Qnn1716) * 30 * 0.065+Qna1715* 20 * 0.1
+B2*1200 * 0.065
+ B3 * 690 * 0.065 + B4 * 690 * 0.065 + B5 * 462 * 0.065
+ waa12+waa21+waa23+waa32+waa34+waa43+waa45+waa54+waa56+waa65
```

```
+waa67 +waa76 +waa78 +waa87 +waa89 +waa98
+waa109 +waa910 +waa1011 +waa1110
+waa1211 +waa1112 +waa1213 +
waa1312 +waa1314 +waa1413 +waa1514 +waa1415;
B1+B2+B3+B4+B5+B6+B7 <= 104+301+750+606
+194+205+201+680+480+300+220+210+420+500;
B1- 800<=0;
B2- 800<=0;
B3- 1000<=0;
B4- 2000<=0;
B5- 2000<=0;
B6- 2000<=0;
B7- 3000<=0;
B1+B2+B3+B4+B5+B6+B7 >= 5171;
B1 = @if(z1#eq#0,0,B1);
B2 = @if(z2#eq#0,0,B2);
B3 = @if(z3#eq#0,0,B3);
B4 = @if(z4#eq#0,0,B4);
B5 = @if(z5#eq#0,0,B5);
B6 = @if(z6#eq#0,0,B6);
B7 = @if(z7#eq#0,0,B7);
500*z1  - B1<=0;
500*z2  - B2<=0;
500*z3  - B3<=0;
500*z4  - B4<=0;
500*z5  - B5<=0;
500*z6  - B6<=0;
500*z7  - B7<=0;
Qnn67 = @if(B1#eq#0,0,Qnn67);
Qnn1514 = @if(B6#eq#0,0,Qnn1514);
Qnn1716 = @if(B7#eq#0,0,Qnn1716);

Qna11 + Qna12 - Qna21 = 0;
Qna21 + Qna23 - Qnn32 = 0;
Qnn32 + Qna34 - Qnn73 = 0;
Qna45 + Qna46 - Qnn54 = 0;
Qnn65 - Qnn54 - Qna57 = 0;
Qnn65 + Qna67 - Qnn76 - B1 + Qnn67 = 0;
Qnn73 + Qnn76 + Qnn78 + Qna78 - B2 - Qnn67 - Qnn87 = 0;
Qnn87 + Qna89 + Qnn89 - B3 - Qnn78 - Qnn98 = 0;
Qnn98 + Qna910 + Qnn910 - Qnn109 - Qnn89 = 0;
Qnn109 + Qnn1012 + Qnn108 - B4 - Qnn910 - Qnn1110 - Qnn1210 = 0;
Qna1111 + Qnn1110 - B5 - Qnn1011 = 0;
Qnn1210 + Qna1212 + Qnn1213 - Qnn1312 - Qnn1012 = 0;
Qnn1312 + Qna1313 + Qnn1314 - Qnn1213 - Qnn1413 = 0;
Qnn1413 + Qnn1416 + Qnn1415 + Qna1414 - Qnn1314 - Qnn1614 - Qnn1514 = 0;
Qna1514 + Qnn1514 - B6 - Qnn1415 = 0;
Qnn1614 + Qna1615 + Qnn1617 - Qnn1416 - Qnn1716 = 0;
Qna1715 + Qnn1716 - Qnn1617 - B7 = 0;

daa12   +  daa21   -  104 = 0 ;
```

```
daa23    +  daa32    -  301 = 0;
daa34    +  daa43    -  750 = 0;
daa45    +  daa54    -  606 = 0;
daa56    +  daa65    -  194 = 0;
daa67    +  daa76    -  205 = 0;
daa78    +  daa87    -  201 = 0;
daa89    +  daa98    -  680 = 0;
daa910   +  daa109   -  480 = 0;
daa1011  +  daa1110  -  300 = 0;
daa1112  +  daa1211  -  220 = 0;
daa1213  +  daa1312  -  210 = 0;
daa1314  +  daa1413  -  420 = 0;
daa1415  +  daa1514  -  500 = 0;

daa12 - Qna11 = 0;
daa21 + daa23 - Qna12 = 0;
daa32 + daa34 - Qna23 = 0;
daa43 + daa45 - Qna34 = 0;
daa54 + daa56 - Qna45 = 0;
daa65 + daa67 - Qna46 = 0;
daa76 + daa78 - Qna57 - Qna67 = 0;
daa87 + daa89 - Qna78 = 0;
daa98 + daa910 - Qna89 = 0;
daa109 + daa1011 - Qna910 = 0;
daa1110 + daa1112 - Qna1111 = 0;
daa1211 + daa1213 - Qna1212 = 0;
daa1312 + daa1314 - Qna1313 = 0;
daa1413 + daa1415 - Qna1514 - Qna1414 = 0;
daa1514 - Qna1715 = 0;


waa12 = @if(daa12#le#1,0.1,0.1*(1+@floor(daa12))*@floor(daa12)/2+0.1);
waa21 = @if(daa21#le#1,0.1,0.1*(1+@floor(daa21))*@floor(daa21)/2+0.1);
waa23 = @if(daa23#le#1,0.1,0.1*(1+@floor(daa23))*@floor(daa23)/2+0.1);
waa32 = @if(daa32#le#1,0.1,0.1*(1+@floor(daa32))*@floor(daa32)/2+0.1);
waa34 = @if(daa34#le#1,0.1,0.1*(1+@floor(daa34))*@floor(daa34)/2+0.1);
waa43 = @if(daa43#le#1,0.1,0.1*(1+@floor(daa43))*@floor(daa43)/2+0.1);
waa45 = @if(daa45#le#1,0.1,0.1*(1+@floor(daa45))*@floor(daa45)/2+0.1);
waa54 = @if(daa54#le#1,0.1,0.1*(1+@floor(daa54))*@floor(daa54)/2+0.1);
waa56 = @if(daa56#le#1,0.1,0.1*(1+@floor(daa56))*@floor(daa56)/2+0.1);
waa65 = @if(daa65#le#1,0.1,0.1*(1+@floor(daa65))*@floor(daa65)/2+0.1);
waa67 = @if(daa67#le#1,0.1,0.1*(1+@floor(daa67))*@floor(daa67)/2+0.1);
waa76 = @if(daa76#le#1,0.1,0.1*(1+@floor(daa76))*@floor(daa76)/2+0.1);
waa78 = @if(daa78#le#1,0.1,0.1*(1+@floor(daa78))*@floor(daa78)/2+0.1);
waa87 = @if(daa87#le#1,0.1,0.1*(1+@floor(daa87))*@floor(daa87)/2+0.1);
waa89 = @if(daa89#le#1,0.1,0.1*(1+@floor(daa89))*@floor(daa89)/2+0.1);
waa98 = @if(daa98#le#1,0.1,0.1*(1+@floor(daa98))*@floor(daa98)/2+0.1);
waa109= @if(daa109#le#1,0.1,0.1*(1+@floor(daa109))*@floor(daa109)/2+0.1);
waa910= @if(daa910#le#1,0.1,0.1*(1+@floor(daa910))*@floor(daa910)/2+0.1);
waa1011= @if(daa1011#le#1,0.1,0.1*(1+@floor(daa1011))
*@floor(daa1011)/2+0.1);
```

```
waa1110= @if(daa1110#le#1,0.1,0.1*(1+@floor(daa1110))
*@floor(daa1110)/2+0.1);
waa1211= @if(daa1211#le#1,0.1,0.1*(1+@floor(daa1211))
*@floor(daa1211)/2+0.1);
waa1112= @if(daa1112#le#1,0.1,0.1*(1+@floor(daa1112))
*@floor(daa1112)/2+0.1);
waa1213= @if(daa1213#le#1,0.1,0.1*(1+@floor(daa1213))
*@floor(daa1213)/2+0.1);
waa1312= @if(daa1312#le#1,0.1,0.1*(1+@floor(daa1312))
*@floor(daa1312)/2+0.1);
waa1314= @if(daa1314#le#1,0.1,0.1*(1+@floor(daa1314))
*@floor(daa1314)/2+0.1);
waa1413= @if(daa1413#le#1,0.1,0.1*(1+@floor(daa1413))
*@floor(daa1413)/2+0.1);
waa1514= @if(daa1514#le#1,0.1,0.1*(1+@floor(daa1514))
*@floor(daa1514)/2+0.1);
waa1415= @if(daa1415#le#1,0.1,0.1*(1+@floor(daa1415))
*@floor(daa1415)/2+0.1);

@bin(z1);
@bin(z2);
@bin(z3);
@bin(z4);
@bin(z5);
@bin(z6);
@bin(z7);
```

A.2.3 Tables

TABLE A.1: The optimal solution of variables

Variables $W_{A_k \leftrightarrow A_{k+1}}$	**Values**
WAA12	108.2000
WAA21	165.4000
WAA23	0.1000000
WAA32	4545.200
WAA34	4336.600
WAA43	10374.10
WAA45	0.1000000
WAA54	18392.20
WAA56	0.1000000
WAA65	1891.600
WAA67	86.20000
WAA76	1336.700
WAA78	37.90000
WAA87	1505.200
WAA89	610.6000
WAA98	16216.60
WAA109	7860.700
WAA910	348.7000
WAA1011	74.20000
WAA1110	3419.200
WAA1211	2.200000
WAA1112	2279.200
WAA1213	2.200000
WAA1312	2070.700
WAA1314	1001.200
WAA1413	3878.200
WAA1514	2702.900
WAA1415	3577.900

TABLE A.2: The optimal solution of variables

Variables d	**Values**
DAA12	46.00000
DAA21	58.00000
DAA23	0.000000
DAA32	301.0000
DAA34	294.1465
DAA43	455.8535
DAA45	0.000000
DAA54	606.0000
DAA56	0.000000
DAA65	194.0000
DAA67	41.94118
DAA76	163.0588
DAA78	27.67421
DAA87	173.3258
DAA89	110.0227
DAA98	569.9773
DAA910	83.43081
DAA109	396.5692
DAA1011	38.25434
DAA1110	261.7457
DAA1112	213.9079
DAA1211	6.092111
DAA1213	6.217028
DAA1312	203.7830
DAA1314	141.0372
DAA1413	278.9628
DAA1415	267.7993
DAA1514	232.2007

TABLE A.3: The optimal solution of variables

Variables $Q_{a,b,j}$	**Values**
QNA11	46.00000
QNA12	58.00000
QNA21	104.0000
QNA23	595.1465
QNN32	699.1465
QNA34	455.8535
QNA45	606.0000
QNA46	235.9412
QNN54	841.9412
QNA57	190.7330
QNN65	1032.674
QNA67	0.000000
QNN76	232.6742
QNN67	0.000000
QNN73	1155.000
QNA78	283.3485
QNN78	0.000000
QNN87	871.0227
QNA89	653.4081
QNN89	0.000000
QNN98	524.4308
QNN108	0.000000
QNA910	434.8235
QNN910	0.000000
QNN109	959.2543
QNA1111	475.6535
QNN1110	959.2543
QNN1011	1434.908
QNA1212	12.30914
QNN1210	0.000000
QNN1012	0.000000
QNN1213	0.000000
QNN1312	12.30914
QNA1313	344.8202
QNN1314	0.000000
QNN1413	357.1293
QNA1414	546.7622
QNN1416	0.000000
QNN1614	0.000000
QNN1415	0.000000
QNN1514	903.8915
QNA1514	0.000000
QNA1615	1434.908
QNN1617	0.000000
QNN1716	1434.908
QNA1715	232.2007

TABLE A.4: The optimal solution of variables

Variables B_i	**Values**
B1	800.0000
B2	800.0000
B3	1000.000
B4	0.000000
B5	0.000000
B6	903.8915
B7	1667.109

TABLE A.5: The optimal solution of variables

Variables Z	**Values**
Z1	1.000000
Z2	1.000000
Z3	1.000000
Z4	0.000000
Z5	0.000000
Z6	1.000000
Z7	1.000000