United International College(UIC)
Zhuhai, Guangdong 2021

The Strategy to Implement an Optimize Pipes Purchasing and Shipping Plan

Beijing Normal University-Hong Kong Baptist University
United International College

Author:
Jiaying YIN 1830006230
Yunzhi XIAO 1830006195
Gengchen SUN 1830021031
Kaiyang LIU 1830004016
Yiran CHEN 1830024027
Yawei YANG 1830025128

Supervisor:
Dr. Ivan Yuhui DENG

Course Title:
Optimization (1001)

Abstract

Pipelines ordering and paving planning are essential for energy resources allocation. We construct a multiple nonlinear optimization model and make a pipes purchasing and shipping plan to get the optimal solution of the total cost. We first use the Floyd-Warshall Algorithm to find the cheapest path between the steel factories and the location points based on the transportation costs on different paths. Then we solve the model. The optimization of pipes purchasing and shipping plan is as follow: We should buy 800 km pipes from Steel factory 1, 800 km pipes from Steel factory 2, 1000 km pipes from Steel factory 3, 1357.35 km pipes from Steel factory 5, and 985.90 km pipes from Steel factory 6. We don't buy pipes from steel factory 4 and 7. The minimal total cost of our model is 1316386.13 units of money (the money unit used is 10,000 Yuan). Our model could be generalized in other similar problems and areas. Some improvements could take into consideration after modifying the model.

Keywords: natural gas pipelines building, nonlinear optimization, the Floyd-Warshall algorithm

Contents

1 Introduction

Natural gas is one of the necessary energy resources in our daily life. However, the locations where natural gas is extracted and where processing plants are located are generally in remote areas and need to be transported over long distances before people can use them. Pipeline transportation is the most common low-cost and convenient way to transport oil and gas resources. It has the advantages of small land area, fast construction, a large volume of oil and gas transportation, high safety performance, low transportation loss, easy management, easy to realize remote centralized monitoring, etc.

Due to the uneven geographical distribution of energy resources, the natural gas pipeline networks in the world are highly integrated networks that move natural gas from factories in gas production areas to storage facilities that are closed to customers. Various planning issues are involved in the laying of pipelines, such as pipeline route design, pipes ordering, and material transportation, and pipeline laying. During the entire pipeline construction process, the government needs to ensure the quality of the pipeline while minimizing construction and maintenance costs.

Our aim for this paper is to propose a pipe purchasing and shipping plan under a specific scenario, such that the governments can finish pipeline laying project with minimum cost. In particular, we only focus on the cost of buying pipes from different factories, the transportation cost from factories to planned pipeline location, and the transportation cost during paving pipes. Our model incorporates the objective cost function and some constraints according to the assumption and the situations in reality.

The remainder of this paper is organized as follows. In Section 2, we show the planning model and give the method and algorithm to solve our model. We present all the numerical results in Section 3 and give the discussion in Section 4. Finally, conclusions are given in Section 5.

2 Mathematical Modeling

2.1 Problem Restatement

The government plans to build a pipeline with 15 location $A_1 \rightarrow A_2 \rightarrow ... \rightarrow A_{15}$. There are seven steel factories S_1, S_2,..., and S_7 can provide the required pipes. The pipes are shipped from factories S_i (i=1,2,...,7) via railway and highway to the location points A_j (j=1,2,..., 15). Pipes are offloaded and paved towards either A_{j-1} direction or A_{j+1} direction. The problem is to propose a pipe purchasing and shipping plan to obtain the minimum total cost for buying and moving pipes.

There are some constraints as follow:

1. All the factories sell at least 500 units pipes.

2. The steel pipe being transported to location A_j can only be used to lay the pipe between location A_{j-1} and location A_{j+1}.

3. All purchased steel pipes are used for laying pipelines.

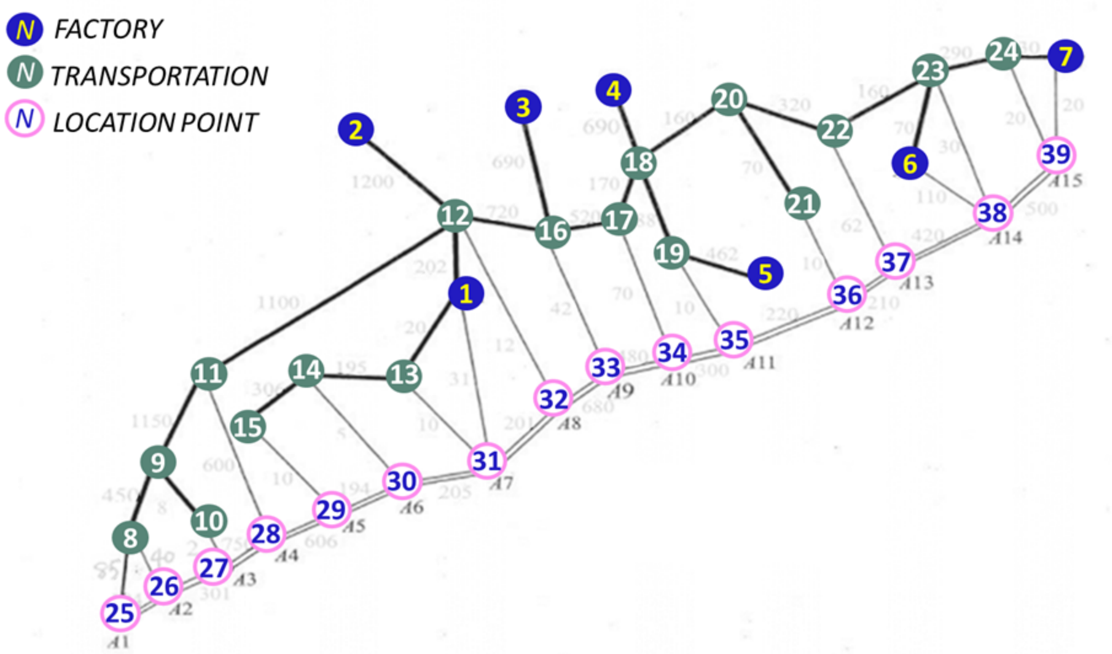

FIGURE 1: Image

2.2 Denotation

Symbols	Descriptions
S_j	The factory that the pipes are bought from, $i = 1, 2, ..., 7$
A_j	The location on the line of delivering natural gas, $j = 1, 2, ..., 15$
K_i	The production capacity of each steel factory S_i
P_i	The price for 1 km (i.e. 1 unit) of pipes for the product of factory S_i
D_j	Distance between location A_j and A_{j+1}
C_{ij}	The minimum transportation cost from factory S_i to location A_j (obtained by the Floyd algorithm previously)
x_{ij}	Number of units (i.e. km) pipes being transported from S_i toward A_j
l_j	Number of units (i.e. km) pipelines being paved from A_j toward A_{j-1}
r_j	Number of units (i.e. km) pipelines being paved from A_j toward A_{j+1}
o_i	The binary variable that decides whether the factory S_i will sell pipes, $O_i = 1$
W_1	The total cost of buying steel pipes
W_2	The total transportation cost for moving pipes between location points
W_3	The total transportation cost for moving pipes from factories to location points
W	The total cost for buying and moving pipes

2.3 Assumptions

In order to better analyze and grasp the essence of the problem, the model has the following reasonable assumptions:

- The factory sells steel pipes with a lower limit of 500 units. In other words, factory S_i won't sell steel pipes unless over 500 units pipe are needed. This is because factories have certain fixed costs to produce and ship products, which can lead to lower profits or even losses if the production volume is too low.
- The pipes shipped to location point A_j are only for paving the interval between A_{j-1} and A_{j+1} which contain A_j. Paving pipes beyond the interval range will lead to higher laying costs and make the cut-off of the location interval meaningless.
- During laying the pipes, transportation costs for a moving distance shorter than 1 km will be counted as 0.1 money unit, which is 1000 Yuan. This is for calculation convenience.
- No pipes shall be wasted during the paving process. It implies that the total amount of pipes required is equal to the length of the entire pipeline $A_1 \to A_2 \to, ..., \to A_{15}$.
- The size and type of steel pipes used in the pipeline from different plants must be consistent to ensure consistent pipeline connections and smooth transportation.
- All steel pipes we need are supplied by the 7 factories $S_i (i = 1, 2, ..., 7)$.
- Assume that the steel pipes are transported without loss on the way.
- Assume that there is no limitation to the number of transportation by highway and railway.

2.4 Modelling

The goal is to minimize the W, which is the total cost for buying and moving pipes (from factories to location points, and between location points).

The source of the total cost can be divided into three parts:

$$W = W_1 + W_2 + W_3$$

The first term W_1 is the transportation cost of moving pipes from factories to location points. There is diverse path choice to transport steel pipes from factories to locations. The cost C_{ij} is the minimum cost among various paths:

$$W_1 = \sum_{i=1}^{7} \sum_{j=1}^{15} C_{ij} x_{ij}$$

The second term W_2 is the transportation cost during paving the pipeline. For every 1km of pipeline laid, the number of steel pipes to be transported is reduced by 1km, and the distance less than 1km is calculated according to the assumption. Therefore, the length of steel pipe transported during the laying process that starts from location A_j can be regarded as summations of arithmetic progression that the first term is l_j and r_j respectively and the last term is 1:

$$W_2 = \frac{0.1}{2}\sum_{j=1}^{15}\left[l_j\left(l_j+1\right)+r_j\left(r_j+1\right)\right]$$

The last term is the cost of buying the pipes from different factories:

$$W_3 = \sum_{i=1}^{7} P_i \sum_{j=1}^{15} x_{ij}$$

Therefore, we can get the objective function:

$$W = \sum_{i=1}^{7}\sum_{j=1}^{15} C_{ij}x_{ij} + \frac{0.1}{2}\sum_{j=1}^{15}\left[l_j\left(l_j+1\right)+r_j\left(r_j+1\right)\right] + \sum_{i=1}^{7} P_i \sum_{j=1}^{15} x_{ij} \tag{1}$$

Subject to:

$$\sum_{i=1}^{7} x_{ij} = l_j + r_j \tag{2}$$

$$l_{j+1} + r_j = D_j (j = 1, ..., 14) \tag{3}$$

$$\sum_{j=1}^{15} x_{ij} \geq 500 o_i \tag{4}$$

$$\sum_{j=1}^{15} x_{ij} \leq K_i o_i \tag{5}$$

$$x_{ij}, l_j, r_j \geq 0$$

$$o_i = 0, 1 \tag{6}$$

$$l_1 = 0, r_{15} = 0 \tag{7}$$

$$i = 1, ..., 7, j = 1, ..., 15$$

The first constraint (2) implies that there no pipes should be wasted. The second constraint (3) requires pipes transported to location A_j will be paved only in the interval $\left[A_{j-1}, A_{j+1}\right]$. The third constraint (4) means every factory will sell pipes only if the order is over 500 units. The inequality constraint (5) restrict the maximum production capacity of each factory. We define variable o_i as the binary variable (6), 0 is for the factory that won't sell pipes because the lower limit of production is not satisfied and 1 is for the factory that will sell pipes. l_1 and $r_1 5$ are set to be zero (7) since according to the given scenario, they are the starting and terminal location of the pipeline.

2.5 Process and algorithm to solve the model

Due to the interlocking of railway and road, it is impossible to directly select the route with the least cost from the steel factory to the location site. So, we adopt the Floyd-Warshall Algorithm here and Compiling it in MATLAB.

Algorithm 1: Floyd-Warshall Algorithm

Input: Total cost matrix
Output: Total cost matrix

```
begin:
Use the Floyd-Warshall Algorithm to find the path with lowest cost
for temp = 1:n do
    for i = 1:n do
        for j = 1:n do
            if (cost of existing path Matrix(i,j) is larger than cost of new path
              Matrix(i,temp) + Matrix(temp,j)) then
                (Matrix(i,j) := cost of new path Matrix(i,temp) + Matrix(temp,j);
            end
        end
    end
end
end
```

We also use MATLAB with the package "fmincon" to solve this nonlinear programming. Because MATLAB can't directly deal with the mix of nonlinear and integer programming. Which means, the constraints (4)(5), where o_i is the 0-1 variables can't work. So we change this constraint to $\sum_{j=1}^{15} x_{ij} \leq K_i$ then solve.

After solving the model without consider the constraint (4), we set the number of production of factories that do not sell any pipes as 0. And find out factories that the number of production is less than 500. Then we set the number of production of these factories to be 0 or 500 one by one to get the optimal solutions which has the minimum cost.

3 Results

I. The transportation cost matrix obtained by Floyd-Warshall algorithm.

II. The cost matrix of the transportation costs plus the materials cost of pipes from distinct factories.

TABLE 1: The table of cost to buy 1km pipes from factory i S_i to location A_j

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
S1	197.13	190.2	160.1	94.475	33.875	14.475	2.3	14.33	64.13	100.73	111.5	120.73	141.73	149.38	167.23
S2	262	255.07	224.97	185.605	125.005	105.605	93.43	79.2	129	165.6	176.37	185.6	206.6	214.25	232.1
S3	275.65	268.72	238.62	199.255	138.655	119.255	107.08	92.85	49.05	85.65	96.42	105.65	126.65	134.3	152.15
S4	320.5	313.57	283.47	244.105	183.505	164.105	151.93	137.7	93.9	62.9	51.57	60.8	81.8	89.45	107.3
S5	311.4	304.47	274.37	235.005	174.405	155.005	142.83	128.6	84.8	53.8	31.03	51.7	72.7	80.35	98.2
S6	321.8	314.87	284.77	245.405	184.805	165.405	153.23	139	95.2	64.2	52.87	41.3	21.15	7.55	25.4
S7	338.05	331.12	301.02	261.655	201.055	181.655	169.48	155.25	111.45	80.45	69.12	57.55	37.4	23.8	2

TABLE 2: The table of cost to buy 1km pipes from factory S_i to location A_j

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
S1	357.13	350.2	320.1	254.475	193.875	174.475	162.3	174.33	224.13	260.73	271.5	280.73	301.73	309.38	327.23
S2	417	410.07	379.97	340.605	280.005	260.605	248.43	234.2	284	320.6	331.37	340.6	361.6	369.25	387.1
S3	430.65	423.72	393.62	354.255	293.655	274.255	262.08	247.85	204.05	240.65	251.42	260.65	281.65	289.3	307.15
S4	480.5	473.57	443.47	404.105	343.505	324.105	311.93	297.7	253.9	222.9	211.57	220.8	241.8	249.45	267.3
S5	466.4	459.47	429.37	390.005	329.405	310.005	297.83	283.6	239.8	208.8	186.03	206.7	227.7	235.35	253.2
S6	471.8	464.87	434.77	395.405	334.805	315.405	303.23	289	245.2	214.2	202.87	191.3	171.15	157.55	175.4
S7	498.05	491.12	461.02	421.655	361.055	341.655	329.48	315.25	271.45	240.45	229.12	217.55	197.4	183.8	162

III. The solution of the model without the limitation of "at least a total of 500km of pipes should be purchased from each factory".

The minimum cost is 1313783.181 units of money (the money unit used is 10,000 Yuan).

TABLE 3: The solution of the model without the limitation of at least 500km

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	Total
S1	0	0	0	255.882	264.428	144.897	134.794	0	0	0	0	0	0	0	0	800
S2	0.92	48.966	208.938	154.524	163.069	43.542	33.441	146.6	0	0	0	0	0	0	0	800
S3	1.022	24.495	184.445	130.033	138.575	19.038	8.961	122.102	371.329	0	0	0	0	0	0	1000
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S5	15.399	13.201	85.787	31.391	39.933	28.148	15.523	23.941	272.681	431.155	400.194	0	0	0	0	1357.35
S6	0	0	0	0	0	0	0	0	0	0	0	87.889	347.754	550.256	0	985.90
S7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	227.747	227.75

IV. The solution of the model with $S_4 = 0$ and $S_7 = 0$.

The minimum cost is 1316386.13 units of money (the money unit used is 10,000 Yuan).

TABLE 4: The solution of the model with $S_4 = 0$ and $S_7 = 0$

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	Total
S1	0	0	0	254.272	262.814	147.831	135.083	0	0	0	0	0	0	0	0	800
S2	3.41	51.643	206.615	152.745	161.286	46.303	33.563	144.434	0	0	0	0	0	0	0	800
S3	4.794	27.421	182.402	128.53	137.072	22.094	9.343	120.221	368.123	0	0	0	0	0	0	1000
S4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S5	9.146	7.586	90.157	36.282	44.825	19.397	14.736	27.995	275.876	431.149	400.201	0	0	0	0	1357.35
S6	0	0	0	0	0	0	0	0	0	0	0	87.899	347.75	617.251	160.749	1213.65
S7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

V. The solution of the model with with $S_4 = 0$ and $S_7 \geq 500$.
The minimum cost is 1319207.09 units of money (the money unit used is 10,000 Yuan).

TABLE 5: The solution of the model with $S_4 = 0$ and $S_7 \geq 500$

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	Total
S_1	0	0	0	253.401	261.946	149.221	135.431	0	0	0	0	0	0	0	0	800
S_2	9.541	48.916	205.277	151.693	160.236	47.513	33.724	143.1	0	0	0	0	0	0	0	800
S_3	1.601	26.118	182.477	128.895	137.436	24.718	10.924	120.303	367.528	0	0	0	0	0	0	1000
S_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S_5	6.207	11.618	91.422	37.841	46.383	14.165	12.645	29.248	276.474	431.149	400.202	0	0	0	0	1357.35
S_6	0	0	0	0	0	0	0	0	0	0	0	80.494	298.766	334.387	0	713.64
S_7	0	0	0	0	0	0	0	0	0	0	0	7.403	48.985	84.611	359.001	500

4 Discussion

4.1 Interpretation of the results

We first solve this problem by ignoring the constraint (4). That is, the number of pipes we bought from the factories can be less than 500. The optimal scheme is that we select the factory S_i which has the least cost for each A_j. According to table 3, the minimum cost is 1313783.181 units of money (the money unit used is 10,000 Yuan). Particularly, number of production of S_4 is 0, and number of production of S_7 is 227.747 units, which is less than 500 units.

We further consider two situations for the factory: whether the factories sell pipes or not. Under this condition, we have two ways to solve it. One is that we let $S_7 = 0$. As table 4 shows, the minimum cost is 1316386.13 units of money (the money unit used is 10,000 Yuan). Another way is we let S_7 is larger or equal to 500. In table 5, $S_7 = 0$, the minimum cost is 1319207.09 units of money (the money unit used is 10,000 Yuan).

Comparing these two solutions, we decide to use the ordering and shipping scheme with factories S_4 and S_7 do not supply any pipes. And the ordering plan are listed in table 4. The total pipes bought from all factories are 5171 units. The total cost is 1316386.13 units of money (the money unit used is 10,000 Yuan).

4.2 Strength and limitation

Our model takes all the requirements into consideration under a specific scenario that is close to reality. The Floyd algorithm we used is one of the simplest algorithms to find the shortest path. In addition, our model has a strong generalization. First of all, the laying of routes and the available traffic roads in the real scenario are far more complex than that scenario, and the pipeline laying directions may not be limited to two. However, all these problems can be modified based on our model to obtain a new model. Second, not only natural gas pipeline laying has such planning problems, but similar planning problems in other fields, such as cable erection and water pipe laying, can be applied to this model.

There are some limitations of our model. We use MATLAB to solve our model, but it is difficult for MATLAB to solve nonlinear programming with binary variables. Therefore we first do not consider the situation that there is no lower limit of production capacity for every factory and get the feasible solution. Then we try all the possible solutions to find the optimal solution. When it comes to a large number of feasible solutions, our method will become inefficient.

4.3 Further discussion

In the real world, there are more complex factors that influence the total cost of pipes problem. Hence, our model should consider the following aspects of the future:

- Production specification restriction

  In this problem, at least a total of 500 km of pipes should be purchased from factories and within maximum production. In solving the problem, we did not consider the limitation of the minimum production volume before and selected the factories that did not meet the requirements in the results and then classified them for discussion.

  If more factories do not meet the production limit in the no constrain results, the solution will be more complicated. Therefore, our model can improve the constrained conditions in the production specification restriction by other software, like Lingo with binary variables.

- The complexity of transportation

  Despite railways and highways, there are other transportation options, such as water transport. Transportation tools like different truck types and railway types also have different prices. Transportation costs in different districts or countries might also vary. The model can add more possibilities for traffic expenses.

- The product loss

  In the model, we assume that no pipes shall be wasted. However, it is inevitable that pipes would be wasted in the welding process and the difference between actual usage and order length.

- Others

  In the production and transportation of pipelines, we also need to consider other factors that might lead to the fluctuation of the total cost, such as costs of labor, and the impact of weather on construction.

5 Conclusion

We propose a nonlinear model to give a pipe ordering and transporting plan by minimizing the total cost during the whole process. To solve the proposed model, We first use the Floyd-Warshall Algorithm to find the least cost paths, then solve the model using MATLAB, and the discussion of the factory which cannot satisfy the minimum production restriction, we constructed the transportation cost matrix and found the equation of the pipe pavement. The final pipe purchasing and shipping plan was showing in table 4, and the total pipes bought from all factories are 5171 units. The minimum cost was 1316386.13 units of money (the money unit used is 10,000 Yuan).

The model considered the production capacity of factories, the ship transportation routes, and the pipe pavement method provided multiple effective solutions for steel pipe purchasing, transportation, and pavement. The minimum cost included the above aspects so that the model achieved the industrial goal of optimizing production volume and the least costly transport route selection to save resources. It can be widely used in the field of engineering construction and material transportation.

Our model is easy to modify and can help the engineer to make the strategy to implement an optimized pipe purchasing and shipping plan in the real world.

The model can be improved in the future. For the production capacity of factories, we can introduce 0-1 variables in the future to represent whether the production volume of each factory reaches the minimum production standard length, and use software such as LINGO to plan and obtain the strategy. In addition, in order to get closer to the actual engineering situation, our model needs to expand the processing capacity of complex transportation, and consider the waste of raw materials in steel pipe welding, labor costs, and even some weather factors.

References

Floyd, Robert W. (June 1962). "Algorithm 97: Shortest Path". *Communications of the ACM*. **5** (6): 345. doi:10.1145/367766.368168.

Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L. (1990). *Introduction to Algorithms* (1st ed.). MIT Press and McGraw-Hill. ISBN 0-262-03141-8. See in particular Section 26.2, "The Floyd–Warshall algorithm", pp. 558–565 and Section 26.4, "A general framework for solving path problems in directed graphs", pp. 570–576.

Luenberger, D. G., Ye, Y. (1984). *Linear and nonlinear programming* (Vol. 2). Reading, MA: Addison-wesley.

Appendix

1. Calculate the distance matrix

```
function Distance_Matrix = initDistanceMatrix(n)

Distance_Matrix = ones(n,n);
for i = 1:39
    for j = 1:39
        if (i == j)
         Distance_Matrix(i,j) = 0;
        else
         Distance_Matrix(i,j) = inf;
        end
    end
end

end
```

2. Calculate the transportation cost matrix

```
function Cost_matrix = transportation_cost()

% Initialization distance matirx of highways
railways_Distance_Matrix = initDistanceMatrix(39);

% Input distances of railway
railways_Distance_Matrix(8,9) = 450;
railways_Distance_Matrix(9,10) = 8;
railways_Distance_Matrix(9,11) = 1150;
railways_Distance_Matrix(11,12) = 1100;
railways_Distance_Matrix(12,1) = 202;
railways_Distance_Matrix(1,13) = 20;
railways_Distance_Matrix(13,14) = 195;
railways_Distance_Matrix(14,15) = 306;
railways_Distance_Matrix(12,2) = 1200;
railways_Distance_Matrix(12,16) = 720;
railways_Distance_Matrix(16,3) = 690;
railways_Distance_Matrix(16,17) = 520;
railways_Distance_Matrix(17,18) = 170;
railways_Distance_Matrix(18,19) = 88;
railways_Distance_Matrix(19,5) = 462;
railways_Distance_Matrix(18,4) = 690;
railways_Distance_Matrix(18,20) = 160;
railways_Distance_Matrix(20,21) = 70;
railways_Distance_Matrix(20,22) = 320;
railways_Distance_Matrix(22,23) = 160;
railways_Distance_Matrix(23,6) = 70;
railways_Distance_Matrix(23,24) = 290;
```

```
railways_Distance_Matrix(24,7) = 30;

for j = 1:39
    for i = 1:j
        if(railways_Distance_Matrix(j,i) > ...
            railways_Distance_Matrix(i,j))
         railways_Distance_Matrix(j,i) = ...
             railways_Distance_Matrix(i,j);
        else
         railways_Distance_Matrix(i,j) = ...
             railways_Distance_Matrix(j,i);
        end
    end
end

railways_Distance_Matrix = Floyd(railways_Distance_Matrix,39);

% Obtain the cost matrix of railways
Cost_railway_Matrix = 0.065 * railways_Distance_Matrix;

% Initialization distance matirx of highways
highways_Distance_Matrix = initDistanceMatrix(39);

% Input distances of highway(transportation part)
highways_Distance_Matrix(8,25) = 85;
highways_Distance_Matrix(8,26) = 40;
highways_Distance_Matrix(10,27) = 2;
highways_Distance_Matrix(11,28) = 600;
highways_Distance_Matrix(15,29) = 10;
highways_Distance_Matrix(14,30) = 5;
highways_Distance_Matrix(13,31) = 10;
highways_Distance_Matrix(1,31) = 31;
highways_Distance_Matrix(12,32) = 12;
highways_Distance_Matrix(16,33) = 42;
highways_Distance_Matrix(17,34) = 70;
highways_Distance_Matrix(19,35) = 10;
highways_Distance_Matrix(21,36) = 10;
highways_Distance_Matrix(22,37) = 62;
highways_Distance_Matrix(6,38) = 110;
highways_Distance_Matrix(23,38) = 30;
highways_Distance_Matrix(24,39) = 20;
highways_Distance_Matrix(7,39) = 20;

% Input distances of highway(paving part)
highways_Distance_Matrix(25,26) = 104;
highways_Distance_Matrix(26,27) = 301;
highways_Distance_Matrix(27,28) = 750;
highways_Distance_Matrix(28,29) = 606;
highways_Distance_Matrix(29,30) = 194;
highways_Distance_Matrix(30,31) = 205;
highways_Distance_Matrix(31,32) = 201;
highways_Distance_Matrix(32,33) = 680;
highways_Distance_Matrix(33,34) = 480;
highways_Distance_Matrix(34,35) = 300;
```

```
highways_Distance_Matrix(35,36) = 220;
highways_Distance_Matrix(36,37) = 210;
highways_Distance_Matrix(37,38) = 420;
highways_Distance_Matrix(38,39) = 500;

for j = 1:39
    for i = 1:j
        if( highways_Distance_Matrix(j,i) > ...
            highways_Distance_Matrix(i,j))
         highways_Distance_Matrix(j,i) = ...
             highways_Distance_Matrix(i,j);
        else
         highways_Distance_Matrix(i,j) = ...
             highways_Distance_Matrix(j,i);
        end
    end
end

highways_Distance_Matrix = Floyd(highways_Distance_Matrix,39);

% Obtain the cost matrix of highways
Cost_highway_Matrix = 0.1 * highways_Distance_Matrix;

% Obtain the Total cost matrix
Total_cost_matrix = ones(39,39);
for i = 1:39
    for j = 1:39
        if(Cost_railway_Matrix(i,j) > Cost_highway_Matrix(i,j))
         Total_cost_matrix(i,j) = Cost_highway_Matrix(i,j);
        else
         Total_cost_matrix(i,j) = Cost_railway_Matrix(i,j);
        end
    end
end

Total_cost_matrix = Floyd(Total_cost_matrix,39);

Cost_matrix = ones(7,15);
for i = 1:7
    for j = 25:39
        Cost_matrix(i,j-24) = Total_cost_matrix(i,j);
    end
end

end
```

3. Code for Floyd

```
function Matrix = Floyd(Matrix,n)

for temp = 1:n
```

```
    for i = 1:n
        for j = 1:n
            if(Matrix(i,j) > Matrix(i,temp) + Matrix(temp,j))
             Matrix(i,j) = Matrix(i,temp) + Matrix(temp,j);
            end
        end
    end
end

end
```

4. Constraints function

```
function  [inequality_constraints, equality_constraints] = ...
    ConstraintsFunc(X_x_r, Cost_matrix, K, D)

% Initialization
% x matrix denotes the pipes transportated from Si to A_j.
x = X_x_r(1:7,1:15);
% r(rj) denotes the length of piplines paving in the right hand ...
    side of Aj
% lj = Dj-1 - rj-1.
r = X_x_r(8,1:15);

% Caluate the sum of xij based on i(1¬7) and j(1¬15) respectively
sum_xij_i = sum(x,2);
sum_xij_j = sum(x,1);

% Construct the inequality constraint with the capacity of ...
    factory Si
inequality_constraints = sum_xij_i - K';
% Construct the inequality constraints about the length of ...
    piplines paving in Aj's right hand side should
% not larger than the distance between Aj and A(j+1) (for j = 1 ¬ 14)
inequality_constraints = [inequality_constraints; (r(1:14) - D)'];

% Construct the equality constraints that the length of piplines ...
    paving in Aj's left hand side
% plus the length of paving piplines in Aj's right hand side ...
    should equal to the total length of
% piplines transpotated in. [denote as Aj = rj + lj = rj + Dj-1 - ...
    rj-1, where Aj is the sum of xij
% based on j]  (for j = 2¬13)
equality_constraints = (sum_xij_j(2:14) - r(2:14) + r(1:13) - ...
    D(1:13))';

% For the A1 and A15, they only have right hand side and left ...
    hand side respectively.
% A1 = r1
equality_constraints(14) = sum_xij_j(1) - r(1);
% r15 = 0
```

```
equality_constraints(15) = r(15);
% The total length of piplines paving should equal to the ...
    distance between A1 and A15.
equality_constraints(16) = sum(sum_xij_i) - sum(D);
% Assume S4 do not sell pipes
equality_constraints(17) = sum_xij_i(4);
% Assume S7 either do not sell pipes or sell 500 pipes, not both
% equality_constraints(18) = sum_xij_i(7);
equality_constraints(18) = sum_xij_i(7)-500;

end
```

5. Objective function

```
function Total_cost = ObjFunc(X_x_r, Cost_matrix, K, D)

% Initialization
% x matrix denotes the pipes transportated from Si to Aj.
x = X_x_r(1:7,1:15);
% r(rj) denotes the length of piplines paving in the right hand ...
    side of Aj
% lj = Dj-1 - rj-1.
r = X_x_r(8,1:15);

% Initialize the function value(total cost)
Total_cost = 0;

% Calculate the cost of the transpotations from Si to Aj and also ...
    the materials cost of factory i.
Total_cost = Total_cost + sum(sum(Cost_matrix .* x));

% Calculate the cost of the transpotations during paving piplines.
% Denote that these terms are the sums of the Arithmetic sequences.
for i=1:14
    Total_cost = Total_cost + ...
        (r(i)*(r(i)+1)+(D(i)-r(i))*(D(i)-r(i)+1)) * 0.05 ;
end

end
```