# 钢管订购和运输的优化

摘要

本文研究的是钢管订购和运输的优化方案，探索了如何通过合理的采购方案和运输途径，在满足各地需求的情况下，花费最低的价钱。针对问题1本文通过首先将连续的需求量离散化，简化问题为几个生产地到几个需求地的问题，之后通过加权的Floyd算法，计算出了各个生产地到各个需求点的最省的运费，最后本文通过非线性规划使用附件B中3.1求解程序，解出了具体的相对省钱的调度方案，并计算出了价格为元。针对问题2，本文又对得到的方案进行分析，通过增加和降低钢厂的生产限额和钢铁售价，利用附件B中3.2求解程序，判断各个厂家价格和产量的变动对最终方案的影响。不仅如此，本文还对这种算法进行扩展，得出更一般路径条件下的最省方案。

**关键词:** 订购和运输 优化 Floyd算法 Lingo 非线性规划 波动影响

# 目录

[摘要 1](#_Toc490548015)

[目录 2](#_Toc490548016)

[符号说明 3](#_Toc490548017)

[1.问题的重述 4](#_Toc490548018)

[2.模型的建立 5](#_Toc490548019)

[2.1 将连续的需求量离散化 5](#_Toc490548020)

[2.2 使用Floyd算法得出最省的运送方案 6](#_Toc490548021)

[2.3 用Lingo程序进行非线性规划 7](#_Toc490548022)

[3. 模型的求解 8](#_Toc490548023)

[3.1 对问题1的计算求解 8](#_Toc490548024)

[3.2 使用MATLAB分析钢厂生产改变所带来的方案波动 8](#_Toc490548025)

[3.3 在更一般道路情况下的推广 9](#_Toc490548026)

[4. 模型的评价 10](#_Toc490548027)

[附录： 11](#_Toc490548028)

[A 参考文献： 11](#_Toc490548029)

[B 程序 11](#_Toc490548030)

[3.1程序：用Lingo求解非线性规划模型 11](#_Toc490548031)

[3.2程序：用MATLAB求解解决方案波动情况 12](#_Toc490548032)

[3.3 程序：用Lingo求解更一般路况下的运送方案 12](#_Toc490548033)

# 符号说明

|  |  |
| --- | --- |
| 各个钢厂 |  |
| 各个天然气管道节点 |  |
| 各个钢厂的产能限制 |  |
| 各个钢厂每单位钢材的售价 |  |
| 节点向左边的钢材运送量 |  |
| 节点向右边的钢材运送量 |  |
| 到的钢材需求量 |  |
| 单位钢材从钢厂到节点的最省的运费(Cost)矩阵 |  |
| 从钢厂到节点的分配量(Amount)矩阵 |  |

# 1.问题的重述

在现实生活中，经常遇到从某些生产地到某些需求地的运输问题，我们总是希望在满足需求的情况下，花费更少的价钱完成任务，这就是本文所需要探究的。现有如下的天然气管道需要足够的钢材，而生产钢材的钢管厂为。它们之间道路的连接图如图1所示：

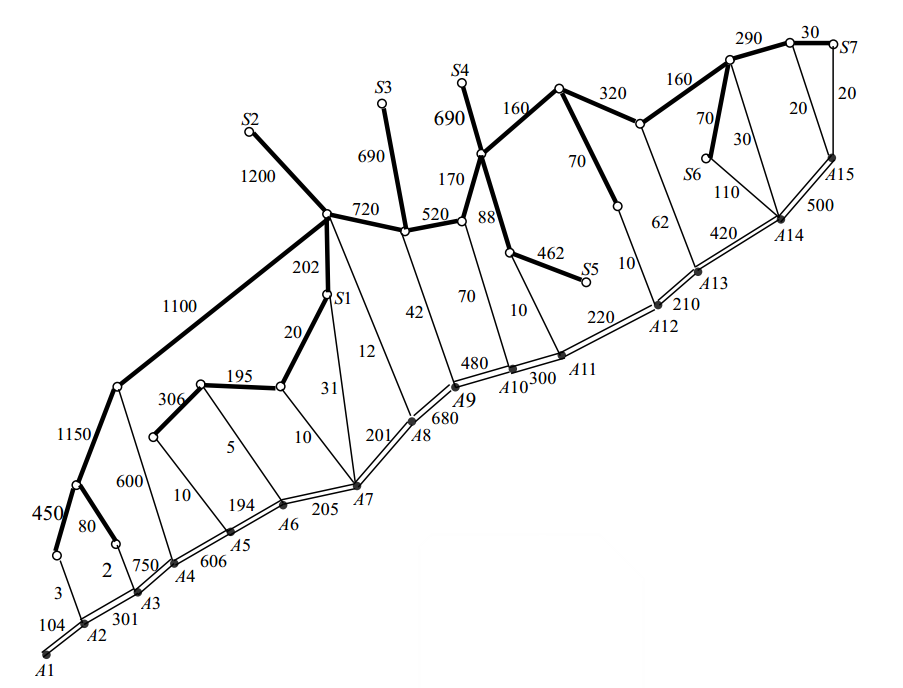


图1：连接各地的公路（细线）和铁路（粗线）网以及管道的具体需求数目

此外，各个钢厂的产量收到一定的限制，各个路段的费用也不同。其中钢厂产量和售价如表1所示，但是每个钢厂要生产就必须生产超过500单位钢材。铁路的费用如表2所示。公路运输费用为 1 单位钢管每公里 0.1 万元。

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | 800 | 800 | 1000 | 2000 | 2000 | 2000 | 3000 |
|  | 160 | 155 | 155 | 160 | 155 | 150 | 160 |

表1 各个钢厂的产量和费用

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 里程(km) | ≤300 | 301～350 | 351～400 | 401～450 | 451～500 |
| 运价(万元) | 20 | 23 | 26 | 29 | 32 |

（续表）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 里程(km) | 501～600 | 601～700 | 701～800 | 801～900 | 901～1000 | 1000以上 |
| 运价(万元) | 37 | 44 | 50 | 55 | 60 | 每增加 1 至 100km 运价增加 5 万元 |

表2 铁路的费用

现在，我们就需要通过这些限制条件，求解出在满足管道钢材需求的情况下，花费最省的钢材购买和运输方案。

# 2.模型的建立

## 2.1 将连续的需求量离散化

根据第1章的描述按照图1，需要钢材的地点是一个连续的路段，这样连续的路段情况复杂，难以得出简洁有效的模型。为了便于计算，我们将需求离散化，即将钢材运送看成两段：

步骤①：钢材由运送到各个节点

步骤②：钢材由各个节点铺到公路上。

步骤②中所需要的费用是便于计算的，这样，我们就把一个连续的问题简化为了一个点对点的需求问题。下面我们先来考虑比较简单的步骤②所需的费用：

根据题意把到的钢材需求量记为，由图1可得关于的数据表3：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  | 104 | 301 | 750 | 606 | 194 | 205 | 201 | 680 |
|  | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
|  | 480 | 300 | 220 | 210 | 420 | 500 | 0 |  |

表3 到的钢材需求量

假设每个节点向左铺的钢材量是，每个节点向左铺的钢材量是，这样就有

特别地，由于节点的左侧和节点的右侧没有管道需求，于是有：

这样我们就可以计算运送到各个需求节点的钢材在步骤②下的运费了，由于每铺下1单位钢材就省1单位的运费，那么铺送个单位的钢材所需要的运费就是

也就是

同理。

于是得到步骤②所需要的所有运费是：

而步骤①所需要的运费需要由Floyd算法得出。

## 2.2 使用Floyd算法得出最省的运送方案

2.1刚才给出了步骤②中的所有运费，现在，我们来考虑步骤①：“钢材由运送到各个节点。”当中的运费。步骤①涉及到公路运费和铁路运费的混合，这里我们使用Floyd算法：

Floyd算法[1]又称为插点法，是一种利用动态规划的思想寻找给定的加权图中多源点之间最短路径的算法。这里我们利用Floyd算法解决最小运费问题，所以权即为对应的路程。由于铁路独特的区别于公路的计费方式，这个问题不再是简单的建立图，加权，求最小值。所以要求得至的最小运费，我们将问题拆解为三块来完成:

a.只通过铁路实现最小运费

b.只通过公路实现最小运费

c.整合A和B的结果,求得公路和铁路混合运费的最小值。

在问题a中，我们规定任意两点间为公路的记为无法通过，给定距离一个极大的权值，使得计算机在求解最小运费时不会通过公路。同理，在问题b中我们将赋予任意两点间为铁路的路段距离一个极大的权值。应用Floyd算法，我们这里以仅以a问题为例。

在问题a中记只通过铁路，任意两点间运费矩阵（行列指标均为，为原图中未标注的点），任意两点间距离矩阵,Floyd算法即转化为对三个指标进行迭代计算 ，再将赋值于就可得到通过Floyd算法求得的只通过铁路的两点最小距离，只需结合铁路公里数运费即可求得对应的任意两点间最小铁路运费矩阵，是一个3939，对角线元素为0的对称阵。注：对于无法通过路段所赋权值（距离）为20000km，在中对应的运费为1010万元( )。

在问题b中,通过类似的方法求得任意两点间公路运费矩阵，无法通过路段在中对应运费为2000万元( )。

在问题c中，先对和中相应指标的每一个元素进行大小比较，取出相同指标的较小值构成一个新矩阵,通过大小比较，原本被规定为不可通过路段由于取了较小值相应的变得可通过了，但也因此破坏了原来的Floyd算法得出的结果，所以我们再次使用Floyd算法，对进行处理，如此，得到单位钢材从钢厂到节点的最省的运费(Cost)矩阵为(与和不同，最省运费矩阵是一个715的矩阵，行指标为，列指标为)。

如此，可以得到单位钢材从钢厂到节点的最省的运费(Cost)矩阵为。

这样，我们就知道各个钢材厂到某个节点运送每单位钢材所需要的最省的运费了，我们再假设从钢厂到节点的分配量(Amount)矩阵为，就可以得到2.1中条件①所需要的运费总和为：

其中根据2.1的条件，我们知道每个节点获得的钢材全用在往左铺和往右铺上，也就是这里的总费用需要加上如下的限制条件：

## 2.3 用Lingo程序进行非线性规划

刚才2.1到2.2详细地分析了运费的问题，现在我们来考虑各个钢厂产量和销售价格对于整个模型的限制：

据题意，每个钢厂的最大生产量为，所以，AMT矩阵中各地对于某个钢厂的需求总和不能超过这个钢厂的最大生产量，同时也不能小于等于500，也就是：

但也有可能某个钢厂根本就不生产，于是上面两个式子还要乘上一个决定某钢铁厂是否要生产的0-1变量，也就是：

如果是1，那么这个钢厂会生产钢铁，如果是0，那么钢厂就不会生产钢铁。

最后我们表达出买钢材所需要的价格为：

现在，我们整理一下解决这个模型所需要的所有的条件：

关于节点的限制

关于钢厂的限制

花费的总额：

我们要求Lingo求解器让上面这个式子有最小值，并且给出此时的分配矩阵。

# 3. 模型的求解

## 3.1 对问题1的计算求解

将2.3末尾的条件和所有已知数据编辑成Lingo程序求解可得：当分配矩阵如下时，花费总额total有最小值

最小值：

Lingo使用的是局部优化求解器。

## 3.2 使用MATLAB分析钢厂生产改变所带来的方案波动

钢厂生产对于最终运送方案的影响主要存在于两个方面：一个方面是对购运计划的影响，另一方面是对总费用total的影响；

对于购运计划的影响，我们用波动过后的分配矩阵和原来的分配矩阵各个对应的元素对应相减并取绝对值后，将新得到的矩阵所有元素相加，就得到了一个变化指数，这个变化指数越大，波动就越大，现在用数学公式表达这个变化因子就是：

而对于总费用的影响，可以直接用波动后的费用减去第一题算得的，即可得到费用差，这个差值可以较好地描述总价格的波动。

为了模拟钢厂生产改变的波动，我们分别把钢厂的生产限额和钢厂的销售价格增加和减少，使用MATLAB计算就可得到如下的处理结果：



根据表格可得以下结论：

的增加对于购运计划影响较大，的增加对于总费用影响较大，的减小对于购运计划影响较大，的减小对于总费用影响较大；

的增加对于购运计划影响较大，的增加对于总费用影响较大，的减小对于购运计划影响较大，的减小对于总费用影响较大。

## 3.3 在更一般道路情况下的推广

新的道路情况如图2所示：

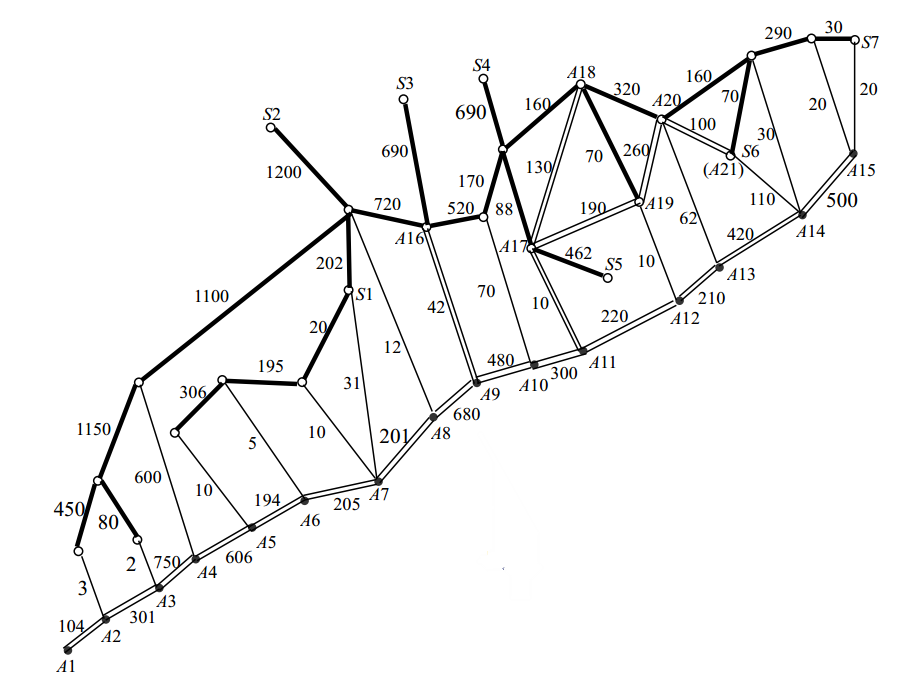


图2 更一般情况下的道路网

同样需要使用Floyd算法单位钢材从钢厂到节点的最省的运费(Cost)矩阵为，只需根据对应的图形，改变在2.1中提到的，，以及相关初值即可求得所需矩阵。

与3.1的方法类似，在利用Floyd算法得出在当前路径下新的分配矩阵后，约束条件是和2.3列出的是一样的。使用Lingo软件得到总费用（total）最小值的运送方式。

最终解得当分配矩阵如下时，花费总额total有最小值：140.6631亿元。

# 4. 模型的评价

本模型通过Lingo、MATLAB和Excel三个软件，给出了钢管订购和运输的比较优化的方案，并且在此基础上分析了钢厂生产改变所带来的方案和总花费波动，最终，我们将我们的方案加以推广，得到了更一般的优化方案。

当然，本方案也有一些局限：我们用Lingo求得的解并不是全局解，这导致了多次运行可能会得到一些不同的结果，不过这些结果相差也不会很大。

# 附录：

## A 参考文献：

[1]百度百科·Floyd算法·https://baike.baidu.com/item/Floyd%E7%AE%97%E6%B3%95/291990?fr=aladdin·2017/8/14

## B 程序

### 3.1程序：用Lingo求解非线性规划模型

MODEL:

TITLE 钢管购运计划;

SETS:

SUPPLY/S1..S7/:S,P,f;

NEED/A1..A15/:everydistance,L,R;

LINK(Supply, need): C, AMT;

ENDSETS

DATA:

S=800 800 1000 2000 2000 2000 3000;

P=160 155 155 160 155 150 144;

everydistance=104 301 750 606 194 205 201 680 480 300 220 210 420 500 0;

C=@text(finalcost.txt);

@TEXT(FinalResult.txt)=@writefor(supply(i):

@writefor(need(j):@format(amt(i,j),'5.0f')), @newline(1) );

@TEXT(FinalFare.txt) =@write(m) ;

ENDDATA

MIN = m;

m = @sum(link(i,j):(C(i,j)+P(i))\*amt(i,j))+@sum(need(j):L(j)^2+L(j)+R(j)^2+R(j))/2\*0.1;

@for(supply(i): @sum(need(j):amt(i,j)) <= S(i)\*f(i) );

@for(supply(i): @sum(need(j):amt(i,j)) >= 500\*f(i));

@for(need(j): @sum(supply(i):amt(i,j)) = L(j)+R(j));

@for(need(j)|j#NE#15: R(j)+L(j+1)=everydistance(j));

L(1)=0;R(15)=0;

@for(supply: @bin(f));

@for(need: @gin(y));

### 3.2程序：用MATLAB求解解决方案波动情况

load init\_fare.txt;

load init\_matrix.txt;

load FinalResult.txt;

load FinalFare.txt;

plused\_matrix = abs(init\_matrix-FinalResult);

solution\_change = sum(sum(plused\_matrix));

result\_change = init\_fare - FinalFare;

C=[C;solution\_change];

R=[R;result\_change]

### 3.3 程序：用Lingo求解更一般路况下的运送方案

MODEL:

TITLE 钢管购运计划;

SETS:

SUPPLY/S1..S7/:S,P,flag;

NEED/A1..A21/:AAMT,l,r,everydistance;

LINK(Supply, need): c, amt;

ENDSETS

DATA:

S=800 800 1000 2000 2000 2000 3000;

P=160 155 155 160 155 150 160;

everydistance=104 301 750 606 194 205 201 680 480 300 220 210 420 500 42 10 130 190 260 100 0;

c=@text(fare.txt);

@TEXT(choice.txt)=@writefor(supply(i):

@writefor(need(j):@format(amt(i,j),'5.0f')), @newline(1) );

ENDDATA

fare1=@sum(supply(i):@sum(need(j): amt(i,j)\*c(i,j)));

fare2=@sum(need(j):(l(j)+1)\*l(j)\*0.1/2)+@sum(need(j):(r(j)+1)\*r(j)\*0.1/2);

fare3=@sum(supply(i):@sum(need(j): amt(i,j)\*p(i)));

totalcost=fare1+fare2+fare3;

MIN=totalcost;

@for(supply(i): @sum(need(j):amt(i,j)) <= S(i)\*flag(i));

@for(supply(i): @sum(need(j):amt(i,j)) >=500\*flag(i));

@for(need(j): @sum(supply(i):amt(i,j))=AAMT(j) );

l(1)=AAMT(1);

@for(need(j)|j#ge#1 #and# j#le#7 :AAMT(j+1)=r(j)+l(j+1));

r(8)+l(9)+l(15)=AAMT(9);

r(9)+l(10)=AAMT(10);

r(10)+l(11)+l(16)=AAMT(11);

@for(need(j)|j#ge#11 #and# j#le#13:r(j)+l(j+1)=AAMT(j+1));

r(14)=AAMT(15);

r(15)=AAMT(16);

r(16)+l(17)+l(18)=AAMT(17);

r(17)=AAMT(18);

@for(need(j)|j#ge#18 #and# j#le#20:r(j)+l(j+1)=AAMT(j+1));

l(21)=0;

r(21)=0;

@for(need(j):l(j)+r(j)=everydistance(j));

@for(need: @gin(AAMT));

@for(need: @gin(amt));

@for(need: @gin(l));

@for(need: @gin(r));

@for(supply: @bin(flag));