2016 高教社杯全国大学生数学建模竞赛

承　诺　书

我们仔细阅读了中国大学生数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们参赛选择的题号是（从 A/B/C/D 中选择一项填写）：

我们的参赛报名号为（如果赛区设置报名号的话）：

所属学校（请填写完整的全名）：重庆邮电大学

参赛队员 (打印并签名) ：1.

2.

3.

指导教师或指导教师组负责人　(打印并签名)：

日期：2015 年 8 月 24 日

赛区评阅编号（由赛区组委会评阅前进行编号）：

2016 高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：

评阅人										
评分										
备注										

全国统一编号（由赛区组委会送交全国前编号）：

全国评阅编号（由全国组委会评阅前进行编号）：

货运列车编组运输问题

摘要

货物运输是一种常见的决策问题，本文通过对运输能力、运输货物的摆放、运输线路的选择三种问题进行充分讨论。

问题一，我们每个车厢货物不同的货物数量为决策变量，车厢的长、高、承重量为约束条件，建立整型规划进行求解，第一次求出其运输货物数量最大的解为 24，第二次求出运输货物量最小的解为 179 吨。

问题二，货物数量限定，那么货物的摆放为对象，使用穷举法求出所有的车厢可行的摆放方式，车厢摆放方式使用次数为决策变量，建立整型规划进行求解。在 B,C,E 三种类型的货物为 68、50、41 件时，最少使用 25 节车厢（I 型车厢使用 13 节，II 型车厢使用 12 节）。在 B,C,E 三种类型的货物为 48、42、52 件时，最少使用 21 节车厢（I 型车厢使用 11 节，II 型车厢使用 10 节）。

问题三，由于已知数据不能确定是否与时间存在必然联系，所以我们对已知数据通过 SPSS 进行统计分析，得出每天上午的集装箱运输需求量服从区间为[90,150]的均匀分布，每天下午的集装箱运输需求量服从均值为 99.99，方差为 20.97834 的正态分布。然后，为解决在每天需求量随机的情况下如何确定列车编组使铁路部门利润最大化问题，建立了以列车编组为自变量的分段函数，表示铁路部门每天的平均利润。然后使用随机模拟的方法求出 F 的最大值，通过反复模拟每天利润的最大值为 51145 元，运输方案为：上午派行一列火车加挂 38 节车厢，下午派行一列火车加挂 37 节车厢。

问题四，针对运输路线优化问题，需要计算运费收入和火车运行成本，由于运费只与集装箱运输数量和站点间最短距离有关。利用 Dijkstra 算法解出 A 到各站点的最短距离。从而得到每个站点的单位运费，而任意站点运往其他站点的集装箱数量，是我们需要求解的数据。决策站台通过哪些路线运输多少集装箱，设立路线的选择，以及集装箱的数量为决策变量，建立线性规划，使用 lingo 软件求解，得到铁路部门利润最大时列车的编组运输方案（详见正文模型第四部分），此种方案下铁路部门利润为 433100 元。

问题五，设计货物运输路线，使得满足各地间运输集装箱的需求条件下，铁路部门利润最大。按照各地间需求量、各线路列车在不同站点实际运输量等变量间七种约束关系，建立整数线性规划模型，求解得到题目要求的列车的编组运输方案（详见正文模型五部分），此种方案下铁路部门利润为 367250 元。

关键词：Dijkstra 算法　　整数规划　　随机模拟　　统计分析

一、 问题重述

问题一：甲地运输五种货物到乙地，由于火车的长度、宽度、高度限制，有许多不同的放置方式，我们需要求出在运输货物数量最多的条件下，运输总重量最小的装运方案。

问题二：甲地运输三种货物到乙地，当货物数量分别为 68、50、41 件情况下求出车厢数量最少的编组方案。若货物数量改为 48,42,52 件，求出车厢数量最少的编组方案。

问题三：从甲地到乙地每天上午和下午各发送一列由Ⅰ型车厢编组的货运列车，货运列车开行成本固定，每增加一个车厢将增加可变成本。为了装卸方便，货物都将放置在长宽高限定的集装箱中，集装箱成本固定。货物需求量是随机的，通过分析附录三中上下午的需求量，试制定两列火车利润最高编组方案。

问题四：铁路部门每天将以 A 站为起点 F 站为终点，沿不同的路线开行若干趟货运列车，每个站点的潜在货物量附录五中可见，全部用Ⅰ型车厢编组，每列火车最大编组量为 40 节车厢。火车开行成本固定。请为铁路部门设计一个利润最大编制运输方案。

问题五：附录六给出了每天各个车站之间潜在的集装箱运输量，铁路部门每天从 A 站用Ⅰ型车厢编组开行到 F 站的若干趟货运列车，铁路网线及费用设定同问题 4，请为铁路部门设计一个编组运输方案。

二、问题分析

对于问题一：

由于题目要求必须按货物的最小长度放置，我们经过分析得出，A，C，D，E 四类货物只能以长度方向放置，B 类货物由于数量的不同，所摆放的方式也不同，所以我们将 B 类货物分成两种 B1 和 B2，当 B 为偶数个时，B 类货物并排摆放，令其组合为 B1。当 B 为奇数个时，先摆放 B1，剩下的一个为 B2。针对上述六种类型的货物长度、重量以及车厢的规格要求，建立约束条件，针对条件目标函数先用 lingo 软件求解最优值，然后将该最优值最为一个约束条件，来计算目标函数的最优解。

对于问题二：

根据题目信息，我们先用 MATLAB 穷举出在三种类型的车厢中所有的放置方法。然后以对各种放置方法的使用次数作为决策变量，解得运输方案的最优解。

对于问题三：

根据题目要求，我们要确定上午下午这两辆列车最佳的编组方案使得利润最大。参考报童模型，我们对附录三给出的最近 100 天上、下午需要运输的集装箱的数量用 SPSS 软件做单样本 K-S 检验，得到其数据所满足的概率分布，并假设

此概率分布无时变。

然后我们采用随机模拟的方法，根据概率分布随机模拟出每天的需求量，然后计算一百年内利润平均值的最大值作为最优解。

对于问题四：

由于集装箱的运费收入是按最短铁路距离来算的，所以我们先用 Djkstra 算法算出 A 站点到其他站点的最短铁路距离，求得单位运费收入。

对于运输成本，分为列车开动与否的固定成本和列车上车厢数目变化的可变成本，因为运输到不同站点、由不同路线运输都会导致成本的变动，所以由不同线路 i 运往不同站点 j 的运量都要分别表示。

目标利润最大=运费收入-固定成本-可变成本。

因为一列火车上最多有 40 节车厢，同时可以通过判断列车是否发行来确定固定成本。这样就得到了目标函数。

可根据 j 路线是否经过 i 站点建立站点 i 与线路 j 之间的关系矩阵 PASS，根据 PASS 矩阵表示运输成本，可以看出这是一个线性规划问题，最后利用 lingo 进行求解。

对于问题五：

我们同样需要先算出运费，采用 floyd 算法计算两两站点之间的最短距离，得出单位运费。

由于在某一路线上运输集装箱时没有限制车厢数量，所以我们只派出一列火车，因此在每条路线上是否有火车开行就成了 0-1 变量问题。

在附录六中，已知每个车站之间潜在的集装箱运输量，可以建立一个 14 个车站到 14 个车站之间的 14 行 14 列运输矩阵，从而每行每列相加，可以得到每一个站点所总共需要发出的集装箱数量和每一个站点所总共需要得到的集装箱数量，也就是每一个站点从所有路线发出和得到的集装箱数量限制。

为表示每一个站点从所有路线发出和得到的集装箱数量，我们建立两个矩阵，一个是 j 站点放到 l 路线上的集装箱数量矩阵 line_on，一个是 j 站点从 l 路线上取下集装箱的数量矩阵 line_up。通过矩阵 line_on 和矩阵 line_up 我们就可以得到每一个站点实际运出和运入的集装箱数量。

三、符号说明

由于本模型所用符号太多，一次说完不容易记忆，所以在模型建立部分分别解释。

四、模型假设

1. 所有货物只能平放。
2. 所有货物都没有优先运输的权利。
3. 集装箱的运费最为铁路部门的收入。
4. 集装箱的需求分布不随时间而变化。

5. 铁路部门的运输成本只考虑题目中所提及的，忽略其他成本。

五、模型的建立与求解

5.1 问题一

5.1.1 模型的准备

针对五种货物运输问题，我们建立整型规划模型，LINGO 求解。其中 B 货物存在两种位置放置的方式，考虑放置方式，我们将其看作两种货物分别 B1、B2，货物重新分类如表一。由于要在运输货物最多的条件下求重量最小，所以我们采用两次优化，第一次求其运输货物最多，第二次求质量最小。

表一 货物重新分类

货物类型	长度（米）	宽度（米）	高度（米）	重量（吨）	数量
A	2.81	3	1.32	5.5	7
B1	2.22	1.5	1.35	10.5	6
B2	1.5	2.22	1.35	21	
C	1.71	3	0.9	9	5
D	2.62	3	1.1	8	7
E	2.53	3	1.2	7.5	6

5.1.2 模型所用符号解释：

符号	含义
x_{ij}	第 i 节车厢中第 j 种货物的数量
$weight_j$	第 j 种货物的重量
l_j	第 i 种货物的长度
$chechang_i$	第 i 节车厢的长度
$chezhong_i$	第 i 节车厢所能承受的最大重量
$huowuliang_j$	第 j 种货物的数量
H_{max}	货物量最大值
W_{min}	货物重量最小值

注：

j=1,2,3,4,5,6 依次表示 A,C,D,E,B1,B2 六种货物。

i=1,2,3,4,5 分别表示 I 型车厢，第一节二型车厢下层，第一节车厢上层，第二节车厢下层，第二节车厢上层。

火车模型如下：

Ⅰ型车厢　Ⅱ型车厢　Ⅱ型车厢

5.1.3 模型的建立

目标函数：

$$H_{max} \mid W_{min}$$

$$H_{max} = \sum_{i=1}^{5} \sum_{j=1}^{6} x_{ij}$$

$$W_{min} = \sum_{i=1}^{5} \sum_{j=1}^{6} x_{ij} \times weight_j$$

约束条件：

1、火车长度的约束：

$$\sum_{j=1}^{6} x_{ij} l_j <= chechang_i$$

2、II 车厢下层剩余长度小于 0.2 米才能装载上层：

$$\sum_{j=1}^{6} x_{2j} l_j >= 14.8$$

$$\sum_{j=1}^{6} x_{4j} * l（j）\geq 14.8 \sum_{j=1}^{6} x_{4j} l_j >= 14.8$$

3、货物重量的约束：

$$\sum_{j=1}^{6} x_{1j} weignt_j <= 55$$

$$\sum_{j=1}^{6} (x_{2j} weignt_j + x_{5j} weignt_j) <= 70$$

$$\sum_{j=1}^{6}(x_{4j}weignt_j + x_{5j}weight_j) <= 70$$

4、货物数量的约束：

$$\sum_{j=1}^{4}\sum_{i=1}^{5} x_{ij} <= huowuliang_j$$

$$\sum_{i=1}^{5}(2x_{i5} + x_{i6}) <= huowuliang_j$$

5、车厢高度的约束：

$$x_{31}=x_{32}=x_{33}=x_{51}=x_{52}=x_{53}=0$$

5.1.4 模型的求解

我们使用 LINGO 软件编程求出其解，使用一次优化求出其最大运输数量为 24，在最大运输数量情况下再次优化，得出其运输总重量最小为 179 吨，运输方案如下：

	A	C	D	E	B
I 型车厢	2	4			
第一节 II 型车厢下层	3			2	1
第一节 II 型车厢上层			3		
第二节 II 车厢下层	2	1		3	
第二节 II 型车厢上层			2	1	

5.2 问题二

5.2.1 模型所用符号解释：

符号	解释
a_i	选用 A 车厢第 i 种放置方式的车厢数
b_i	选用 B 车厢第 i 种放置方式的车厢数
c_i	选用 C 车厢第 i 种放置方式的车厢数
n_{ijk}	k 类车厢第 i 种方式中放置 j 货物的数目
l_j	j 货物的长度
w_j	j 货物的重量
W_k	第 k 类车厢的承重上限
L_k	第 k 类车厢的长度

5.2.2 模型的建立

第一步：我们把车厢分为三类：I 型车厢 A、单装下层的 II 型车厢 B、两层都装的 II 型车厢 C。

第二步：用 matlab 穷举出所有车厢所有的装运方式，用矩阵 A、B、C 表示。

第三步：令发行的车厢总数为 Min，则由题意得：

$$\mathrm{Min} = \sum a_i + \sum b_i + \sum c_i$$

约束条件：

1、每个车厢放置的货物不超重：

$$\sum_j n_{ijk} w_j <= W_k$$

2、每个车厢放置的货物不超长：

$$\sum_j n_{ijk} l_j <= L_k$$

3、B 类车厢最多只能用一节：

$$\sum b_i = 1$$

5.2.3 模型的求解

由 matlab 解得：

A、B、C 三类车厢的放置方式分别有 91、176、279 种。

当 B,C,E 三种类型的货物各 68、50、41 件时，由 lingo 解得：

最少使用 25 节车厢，详细放置方式如下：

<table>
<tr><td>装运方式</td><td>使用数量</td><td>车厢类型</td><td>B1</td><td>B2</td><td>C</td><td>E</td></tr>
<tr><td>a40</td><td>3</td><td>I 型车厢</td><td>0</td><td>1</td><td>4</td><td>1</td></tr>
<tr><td>a56</td><td>10</td><td>I 型车厢</td><td>0</td><td>3</td><td>0</td><td>3</td></tr>
<tr><td rowspan="2">c146</td><td rowspan="2">4</td><td>II 型车厢(下)</td><td>0</td><td>1</td><td>3</td><td>2</td></tr>
<tr><td>II 型车厢(上)</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td rowspan="2">c272</td><td rowspan="2">8</td><td>II 型车厢(下)</td><td>1</td><td>2</td><td>3</td><td>0</td></tr>
<tr><td>II 型车厢(上)</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

当 B,C,E 三种类型的货物各 48、42、52 件时，由 lingo 解得：

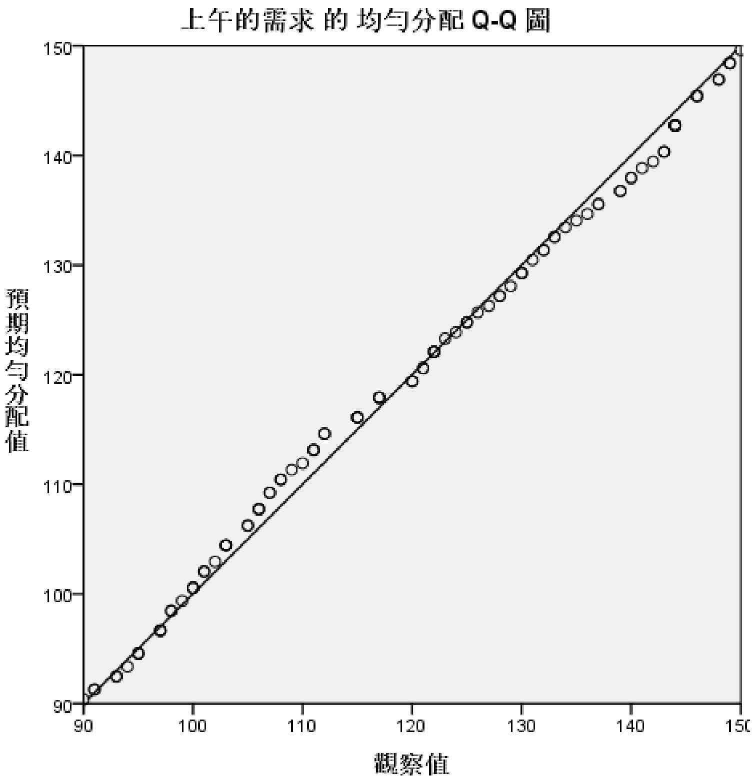
最少使用 21 节车厢，详细放置方式如下：

<table>
<tr><td>装运方式</td><td>使用数量</td><td>车厢类型</td><td>B1</td><td>B2</td><td>C</td><td>E</td></tr>
<tr><td>a40</td><td>5</td><td>I 型车厢</td><td>0</td><td>1</td><td>4</td><td>1</td></tr>
<tr><td>a56</td><td>6</td><td>I 型车厢</td><td>0</td><td>3</td><td>0</td><td>3</td></tr>
<tr><td rowspan="2">c22</td><td rowspan="2">1</td><td>II 型车厢(下)</td><td>0</td><td>0</td><td>0</td><td>5</td></tr>
<tr><td>II 型车厢(上)</td><td>0</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td rowspan="2">c34</td><td rowspan="2">2</td><td>II 型车厢(下)</td><td>0</td><td>0</td><td>1</td><td>4</td></tr>
<tr><td>II 型车厢(上)</td><td>0</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td rowspan="2">c79</td><td rowspan="2">1</td><td>II 型车厢(下)</td><td>0</td><td>0</td><td>3</td><td>3</td></tr>
<tr><td>II 型车厢(上)</td><td>0</td><td>0</td><td>2</td><td>0</td></tr>
<tr><td rowspan="2">c224</td><td rowspan="2">1</td><td>II 型车厢(下)</td><td>0</td><td>5</td><td>0</td><td>1</td></tr>
<tr><td>II 型车厢(上)</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td rowspan="2">c272</td><td rowspan="2">5</td><td>II 型车厢(下)</td><td>1</td><td>2</td><td>3</td><td>0</td></tr>
<tr><td>II 型车厢(上)</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

5.3 问题三

5.3.1 模型的建立

第一步：对近一百天上午、下午需要运输的集装箱数量用 SPSS 做出 Q-Q 图以及单样本 K-S 检验，分析其满足的分布属性：



图一 上午需求量的均匀分布 Q-Q 图

單一樣本 Kolmogorov-Smirnov 檢定

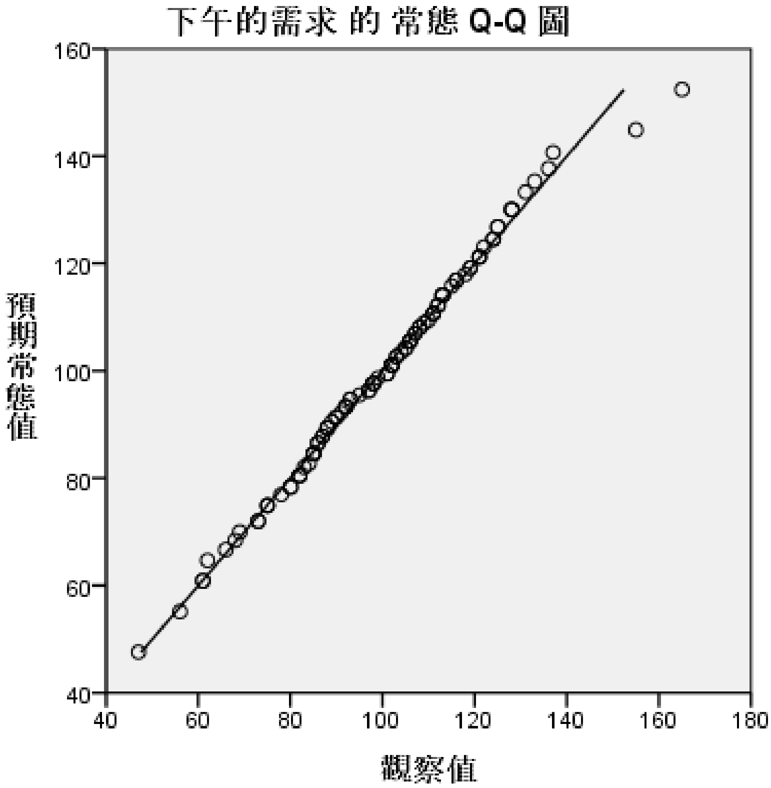
		上午的需求
N		100
均勻參數 a,b	最小值	90
	最大值	150
最極端差異	絕對	.053
	正	.053
	負	-.053
Kolmogorov-Smirnov Z 檢定		.533
漸近顯著性 （雙尾）		.939
精確顯著性（雙尾）		.924
點機率		.000

a. 檢定分配是均勻的。

b. 從資料計算。

图二 上午需求量的单样本 K-S 检验

通过分析，上午需求量对平均分布的显著性接近 1，所以我们可以认为每天上午的集装箱需求量满足均匀分布 U[90,150]。



图三 下午需求量的正态分布 Q-Q 图

單一樣本 Kolmogorov-Smirnov 檢定

		下午的需求
N		100
常態參數 [a,b]	平均數	99.99
	標準偏差	20.978
最極端差異	絕對	.041
	正	.041
	負	-.039
測試統計資料		.041
漸近顯著性 （雙尾）		.200[c,d]
精確顯著性（雙尾）		.994
點機率		.000

a. 檢定分配是常態的。

b. 從資料計算。

c. Lilliefors 顯著更正。

d. 這是 true 顯著的下限。

图二 下午需求量的单样本 K-S 检验

通过分析，下午需求量对正态分布的显著性接近 1，所以我们可以认为每天下午的集装箱需求量满足正态分布 N(99.99，20.97834)。

第二步：假设每天上午的需求为 y1,下午的需求为 y2。其中 y1 满足 U[90,150]，y2 满足 N(99.99，20.97834)。假设每天上午开行加挂 x1 节车厢，每天下午开行加挂 x2 节车厢。令每天的利润为 F，根据题意，则有：

$$F=\begin{cases}160000-1500(x1+x2)\ldots\ldots\ldots\ldots\ldots 3x1>Y1,3x2>Y2\\154000-1500x2-1350x1\ldots\ldots\ldots\ldots 3x1<Y1,3x1+3x2>Y1+Y2\\40000-1500x1+2100x2\ldots\ldots\ldots\ldots 3x1>Y1,3x2<Y2\\-110000+2250x1+2100x2\ldots\ldots\ldots 3x1<Y1,3x1+3x2<Y1+Y2\end{cases}$$

5.3.2 模型的求解

用随机模拟的方法求出 F 的最大值。

我们根据 y1、y2 的分布，用 matlab 先模拟了 100 年的数据，然后求取这一百年内 F 平均值的最大值及其相对应的 x1、x2 的值。反复模拟 5 次，模拟结果如下：

模拟年数/年	F 的最大值	X1	X2
100	51145	38	37
100	51145	38	37
100	51145	38	37
100	51145	38	37
100	51151	38	37

从模拟结果可知，每天利润的最大值为 51145 元，运输方案为：上午派行一列火车加挂 38 节车厢，下午派行一列火车加挂 37 节车厢。

5.4 问题四

5.4.1 模型所用符号解释：

符号	含义
s_i	第 i 条路线的总路程
$send_{ij}$	第 i 条路线运往 j 站点的集装箱数
n_i	第 i 条路线开行的车厢数
d_i	A 站点到第 i 个站点的最短铁路距离
$fare_i$	单位货物运送到第 i 个站点的单位路程运费

$need_j$	j 站点的需求
$pass_{ij}$	第 i 条路线与 j 站点的连接关系，1 表示连接

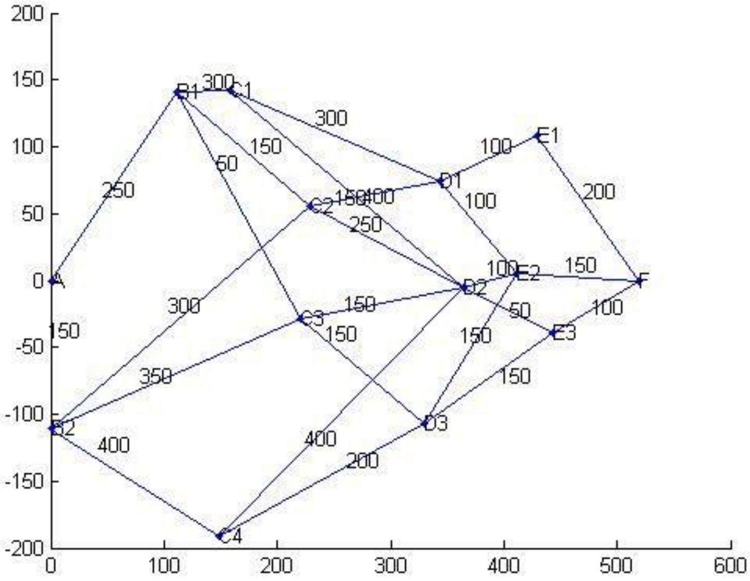
5.4.2 模型的建立

第一步：先用 Dijkstra 算法得到站点 A 到其他每个站点的最短铁路距离，从而得到每个站点的单位运费。

第二步：例举出每条路线，共有 24 条路线与 14 个站点，然后给出路线与站点之间的关系矩阵 PASS, $pass_{ij}$=1 表示第 i 条路线经过 j 站点，0 表示不经过。每条线路经过的站台，见表二。各站点连接图，见图一。

表二 详细路线

线路	经过站台	线路	经过站台
1	AB1C1D1E1F	13	AB2C2D1E1F
2	AB1C1D1E2F	14	AB2C2D1E2F
3	AB1C1D2E3F	15	AB2C2D2E2F
4	AB1C1D2E3F	16	AB2C2D2E3F
5	AB1C2D1E1F	17	AB2C3D2E2F
6	AB1C2D1E2F	18	AB2C3D2E3F
7	AB1C2D2E2F	19	AB2C3D3E2F
8	AB1C2D2E3F	20	AB2C3D3E3F
9	AB1C3D2E2F	21	AB2C4D2E2F
10	AB1C3D2E3F	22	AB2C4D2E3F
11	AB1C3D3E2F	23	AB2C4D3E2F
12	AB1C3D3E3F	24	AB2C4D3E3F



图一 各站点连接图

第三步：令利润为 Y，则由题意得：

$$Y = 2\sum_{j=1}^{14} s_i \sum_{i=1}^{24} \mathrm{send}_{ij} - 15000\sum_{i=1}^{24}\left[\frac{n_i}{40}\right]_+ - \sum_{i=1}^{24} d_i \mathrm{fare}_i$$

约束条件：

1、如果 pass_{ij} =0,那么 send_{ij} =0

2、运往 j 站点的货物不超过其需求

$$\sum_{i=1}^{24} \mathrm{send}_{ij} <= \mathrm{need}_j$$

3、除了最后一个车厢，每个车型都必须装满

$$3(n_i - 1) <= \sum_{i=1}^{14} \mathrm{send}_{ij} <= 3n_i$$

5.4.3 模型的求解

用 lingo 进行求解，解得：

最大利润为 433100 元，其相应的运输方案如下：

路线	n2	n5	n6	n7	n8	n18	n24
车厢	27	8	112	21	23	5	42

路线 2 开行一列火车，编组 27 节车厢；
路线 5 开行一列火车，编组 8 节车厢；
路线 6 开行三列火车，前两列编组 40 节车厢，第三列编组 32 节车厢；
路线 7 开行一列火车，编组 21 节车厢；
路线 8 开行一列火车，编组 23 节车厢；
路线 18 开行一列火车，编组 5 节车厢；
路线 24 开行两列火车，第一列编组 40 节车厢，第二列编组 2 节车厢。

派送方案如下：

send_{ij}	1	2	3	4	5	6	7	8	9	10	11	12	13	14
n2		1		80										
n5					1			1			22			
n6		57			53			81				72		72
n7									63					
n8													69	
n18			1			14								
n24			1				71			54				

注：1~14 表示站点 A~F，空白区域默认为 0

5.5 问题五

5.5.1 模型所用符号解释

符号	解释
$load_{ij}$	i 站点运往 j 站点的集装箱数
$load\_l_{ijl}$	i 站点通过路线 l 运往 j 站点的集装箱数
$need_{ij}$	i 站点运往 j 站点的潜在需要集装箱数
$line\_on_{lj}$	j 站点上传 l 线路的集装箱数
$line\_up_{lj}$	j 站点下载 l 线路的集装箱数
$line\_max_{l}$	第 l 条线路运输的最大集装箱
S_{ij}	i、j 两个站点的最短距离
L_{l}	第 l 条路线的总长度
go_{l}	第 l 条线路是否发车，1 表示发车
$pass_{lj}$	第 l 条线路是否经过 j 站点，1 表示经过
$xiang_{l}$	l 路线上托运的车厢数

5.5.2 模型的建立

由于第四问已经得出了路线与站点的连接关系 $pass_{ij}$，此处不再重复。

第一步：用 Floyd 算法算出两两站点之间的最短铁路距离，用矩阵 S_{ij} 表示。

第二步：由于没有限制车厢数量，所以每条线路最多发一列火车利润最大。

第三步：令利润为 G，则由题意得：

$$G = -15000\sum_{l} go_{l} - \sum_{l} xiang_{l} \times L_{l} + \sum_{j}\sum_{i} load_{ij} \times S_{ij} \times 2$$

约束条件：

1、每个站点运输不超过潜在运输：

$$load_{ij} <= need_{ij}$$

2、j 站点上传 l 路线的货物等于 j 站点通过 l 对其他所有站点的运输货物总和：

$$line\_on_{lj} = pass_{lj} \times (\sum_{i} load\_l_{jil} \times pass_{li})$$

3、j 站点下载 l 路线的货物等于所有其他站点通过 l 对 j 站点运输的货物总和：

$$line\_up_{lj} = pass_{lj} \times (\sum_{i} load\_l_{ijl} \times pass_{li})$$

4、i 通过所有路线运往 j 的货物数量之和就等于 i 实际运往 j 的货物数量：

$$load_{ij} = \sum_{l} load\_l_{ijl}$$

5、第 l 条路线运送的最大货物数为运输过程中的最大货物数：

$$line\_max_{l} = \max\{\sum_{j=1}^{1}(line\_on_{lj} - line\_up_{lj}), \sum_{j=1}^{2}(line\_on_{lj} - line\_up_{lj}), \ldots, \sum_{j=1}^{14}(line\_on_{lj} - line\_up_{lj})\}$$

6、每条路线的上传总量等于下载总量：

$$\sum_{j} line\_on_{lj} = \sum_{j} line\_up_{lj}$$

7、每条线运输货物最大量不超过车厢数的三倍：

$$line\_max_{l} <= 3 \times chexiang_{l}$$

5.5.3 模型的求解

用 lingo 进行求解，解得其，最大利润为 367250，其相应的运输方案为：

第四条线路发行一列火车，加挂 55 节车厢；

第十三条线路发行一列火车，加挂 40 节车厢；

第十六条线路发行一列火车，加挂 31 节车厢；

第二十一条线路发行一列火车，加挂 43 节车厢。

总体运送方案如下：

A(i，j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1				10									44	56
2									71					
3											99	57	49	15
4													93	
5											19			
9													4	

注：

A(i ， j)为第 i 个站点运往第 j 个站点的集装箱个数，空白处默认为零。

每条路线的运输方案如下：

L4:

L4(i,j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1				10										
2									71					
4													93	
9													4	

L13:

L13(i,j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3											99			
4														
5											19			

L16:

L16(i,j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1													44	
3													49	

L21:

L21(i,j)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1														56
3												57		15

六、模型的评价与改进

6.1 模型的评价

本文所阐建立的模型都是基于题目已有的数据，对数据进行分析处理。第一问到第五问基本上都是整数线性规划的模型。其中，第三问我们通过处理附录所给的集装箱需求数据，得出了每天的集装箱需求分布，然后随机模拟，我们加大了模拟的力度，模拟了 100 年的数据，所以在集装箱需求分别不变的情况下，得出的结果是非常逼近真实的最优值的。

6.2 模型的改进

1、本模型只是针对该地点的铁路部门运输情况而建立的，所以不容易移植，我们可以通过加大变量数量，把本模型所有与地点、需求、费用、路线分布等常量因素变成变量加入到模型中，然后建立输入函数以便此模型适用于各种时变环境。

2、本题所涉及的所有货物都一视同仁，没有优先运输权、没有保质期等等普通货物应该拥有的特征，所以我们可以对货物赋予权重，在建模时考虑各种权重因素以便此模型适用于实际的货物运输。

3、我们在计算铁路局收入的时候没有考虑题外的成本，其实铁路局的成本远远不止这些，铁路的损耗、火车的维护、驾驶员的工资、铁路路线成本（或租金）等等，所以我们可以把这些因子加入到此模型中，让此模型不仅输出与运输方案有关的决策，还与时间、天气、铁路轨道质量等等因素有关。

参考文献

[1].姜启源，谢金星，叶俊，数学模型（第三版），北京：高等教育出版社，2003.8；

附录

附录一：模型所用代码

第一问：

```
sets:
huowu/A C D E B1 B2/:length,height,weight;
che/1..5/:long;
link(huowu,che):x;
endsets

data:
length=2.81 1.71 2.62 2.53 2.22 1.5;
  height=1.32 0.90 1.10 1.20 1.35 1.35;
weight=5.5 9 8 7.5 21 10.5;
long=12.5 15 15 15 15;
  enddata

!min=@sum(huowu(i):@sum(che(j):weight(i)*x(i,j)));
max=@sum(huowu(i):@sum(che(j):x(i,j)))+@sum(che(j):x(5,j));

@for(che(j):@sum(huowu(i):length(i)*x(i,j))<=long(j));
@sum(huowu(i):length(i)*x(i,2))>14.8;
@sum(huowu(i):length(i)*x(i,4))>14.8;
@sum(huowu(i):weight(i)*x(i,1))<55;
@sum(huowu(i):weight(i)*x(i,2)+weight(i)*x(i,3))<70;
@sum(huowu(i):weight(i)*x(i,4)+weight(i)*x(i,5))<70;

@sum(che(j):x(1,j))<=7;
@sum(che(j):x(2,j))<=5;
@sum(che(j):x(3,j))<=7;
```

```
@sum(che(j):x(4,j))<=6;
@sum(che(j):2*x(5,j)+x(6,j))<=7;

@for(huowu(i):@for(che(j):@gin(x(i,j))));
x(1,3)=0;
x(1,5)=0;
x(5,3)=0;
x(5,5)=0;
x(6,3)=0;
x(6,5)=0;
```

第二问：

1、matlab 穷举放置方式矩阵程序：

```
%把车厢分为 I II 只装下层  II 双层  A 为 I 的装载可行矩阵，B 为 II 只装下层的可行矩阵，C 为双层的可行矩阵
%
w=[21 10.5 9 7.5];          %B1 B2 C E  重量
l=[2.22 1.5 1.71 2.53]; %B1 B2 C E  长度
i=1;
j=1;
k=1;
for n11=0:5                     %nij 表示第 i 个货物装在第 j 车厢的个数
    for n21=0:8
        for n31=0:7
            for n41=0:4
                if( n11*w(1) + n21*w(2) + n31*w(3) + n41*w(4) <= 55 && n11*l(1) + n21*l(2) + n31*l(3) + n41*l(4) <= 12.5 )
                    A(i,:) = [n11 n21 n31 n41];
                    i=i+1;
                end
            end
        end
    end
end

for n12=0:6                     %nij 表示第 i 个货物装在第 j 车厢的个数
    for n22=0:10
        for n32=0:8
            for n42=0:5
                if( n12*w(1) + n22*w(2) + n32*w(3) + n42*w(4) <= 70 && n12*l(1) + n22*l(2) + n32*l(3) + n42*l(4) <= 10 )
                    B(j,:) = [n12 n22 n32 n42];
                    j=j+1;
                end
            end
        end
    end
```

```
end

for n13=0:6                    %nij 表示第 i 个货物装在第 j 车厢的下层个数      m 表示上层
    for n23=0:10
        for n33=0:8
            for n43=0:5
                for m1=0
                    for m2=0
                        for m3=0:8
                            for m4=0:5
                                if( (n13+m1)*w(1) + (n23+m2)*w(2) + (n33+m3)*w(3) + (n43+m4)*w(4) <= 70 && n13*l(1) + n23*l(2) + n33*l(3) + n43*l(4) <= 15 && n13*l(1) + n23*l(2) + n33*l(3) + n43*l(4) >= 10 && m1*l(1) + m2*l(2) + m3*l(3) + m4*l(4) <= 15)
                                    C(k,:) = [n13 n23 n33 n43 m1 m2 m3 m4];
                                    k=k+1;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
```

2、lingo 求解线性规划程序：

```
sets:
aa/1..91/:a;
bb/1..176/:b;
cc/1..279/:c;
huo_x/1 2 3 4/:;
huo_y/1..8/:;
link1(aa,huo_x):AAA;
link2(bb,huo_x):BBB;
link3(cc,huo_y):CCC;
endsets
data:
AAA=@OLE('D:\data_A.xls',AAA);
BBB=@OLE('D:\data_B.xls',BBB);
CCC=@OLE('D:\data_C.xls',CCC);

enddata
min=@sum(aa(i):a(i))+@sum(bb(i):b(i))+@sum(cc(i):c(i));

2*(@sum(aa(i):-a(i)*AAA(i,1))+@sum(bb(i):-b(i)*BBB(i,1))+@sum(cc(i):-c(i)*(CCC(i,1)+CCC(i,5))))+@sum(aa(i):-
```

```
a(i)*AAA(i,2))+@sum(bb(i):-b(i)*BBB(i,2))+@sum(cc(i):-c(i)* (CCC(i,2)+CCC(i,6)))<-48;
@sum(aa(i):-a(i)*AAA(i,3))+@sum(bb(i):-b(i)*(BBB(i,3)))+@sum(cc(i):-c(i)*(CCC(i,3)+CCC(i,7)))<-42;
@sum(aa(i):-a(i)*AAA(i,4))+@sum(bb(i):-b(i)*(BBB(i,4)))+@sum(cc(i):-c(i)*(CCC(i,4)+CCC(i,8)))<-52;
@sum(bb(i):b(i))<=1;
@sum(cc(i):c(i))+@sum(bb(i):b(i))<=@sum(aa(i):a(i))-1;
@for(aa:@gin(a));
@for(bb:@gin(b));
@for(cc:@gin(c));
```

第三问：

随机模拟算法 matlab 程序：

```
clear;
i=1;
m=1;
n=1;
l=1;
B=[0 0 0];

for kk=1:5

        Y11 = unifrnd(90,150,1,36500);
        Y22 = normrnd(99.99,20.97834,1,36500);
        Y11 = round(Y11);Y11(1,1)
        Y22 = round(Y22);Y22(1,1)
               for x1=1:50
                  for x2=1:60
                    sum=0;
                    for m=1:36500
                       Y1=Y11(m);
                       Y2=Y22(m);
                       if 3*x1>Y1 && 3*x2>Y2
                          y = 160000-1500*(x1+x2);
                       elseif 3*x1<Y1 && 3*x1+3*x2>Y1+Y2
                          y = 154000-1500*x2-1350*x1;
                       elseif 3*x1>Y1 && 3*x2<Y2
                          y = 40000-1500*x1+2100*x2;
                       elseif 3*x1<Y1 && 3*x1+3*x2<Y1+Y2
                          y = -110000+2250*x1+2100*x2;
                       end
                       sum=sum+y;
                    end
                    A(i,:)=[x1,x2,sum/36500];
                    if B(3)<A(i,3)
                       B = A(i,:);
```

```
                    end
                    i=i+1;

                  end
               end
       D(l,:)= B;
       l=l+1;
end
```

第四问：

1、Dijk 算法 matlab 程序：

```
function [ distance path] = Dijk( W,st,e )
%DIJK Summary of this function goes here
%    W   权值矩阵    st 搜索的起点    e 搜索的终点
n=length(W);%节点数
D = W(st,:);
visit= ones(1,n); visit(st)=0;
parent = zeros(1,n);%记录每个节点的上一个节点

path =[];

for i=1:n-1
    temp = [];
    %从起点出发，找最短距离的下一个点，每次不会重复原来的轨迹，设置 visit 判断节点是否访问
    for j=1:n
       if visit(j)
           temp =[temp D(j)];
       else
           temp =[temp inf];
       end

    end

    [value,index] = min(temp);

    visit(index) = 0;

    %更新 如果经过 index 节点，从起点到每个节点的路径长度更小，则更新，记录前趋节点，方便后面回溯循迹
    for k=1:n
        if D(k)>D(index)+W(index,k)
           D(k) = D(index)+W(index,k);
           parent(k) = index;
        end
    end
```

```
end
distance = D(e);%最短距离
%回溯法    从尾部往前寻找搜索路径
t = e;
while t~=st && t>0
  path =[t,path];
  p=parent(t);t=p;
end
path =[st,path];%最短路径
end
```

2、matlab 发车路线以及求解路线_站点关系程序：

```
clear;
M=[0 250 150 inf inf inf inf inf inf inf inf inf inf inf
     250 0    inf 50 150 300 inf inf inf inf inf inf inf inf
     150 inf 0 inf 400 350 300 inf inf inf inf inf inf inf
     inf 50 inf 0 inf inf inf 300 400 inf inf inf inf inf
     inf 150 400 inf 0 inf inf 150 250 inf inf inf inf inf
     inf 300 350 inf inf 0 inf inf 150 250 inf inf inf inf
     inf inf 300 inf inf inf 0 inf 400 200 inf inf inf inf
     inf inf inf 300 150 inf inf 0 inf inf 100 100 inf inf
     inf inf inf 400 250 150 400 inf 0 inf inf 50 100 inf
     inf inf inf inf inf 150 200 inf inf 0 inf 150 150 inf
     inf inf inf inf inf inf inf 100 inf inf 0 inf inf 200
     inf inf inf inf inf inf inf 100 50 150 inf 0 inf 150
     inf inf inf inf inf inf inf inf 100 150 inf inf 0 100
     inf inf inf inf inf inf inf inf inf inf 200 150 100 0];
for i=1:14
       [yunfei(i) path] = Dijk(M,1,i);
end
N=[0 250 150 0 0 0 0 0 0 0 0 0 0 0            %A
     0 0 0 50 150 300 0 0 0 0 0 0 0 0          %B1
     0 0 0 0 400 350 300 0 0 0 0 0 0 0         %B2
     0 0 0 0 0 0 0 300 400 0 0 0 0 0           %C1
     0 0 0 0 0 0 0 150 250 0 0 0 0 0           %C2
     0 0 0 0 0 0 0 0 150 250 0 0 0 0           %C3
     0 0 0 0 0 0 0 0 400 200 0 0 0 0           %C4
     0 0 0 0 0 0 0 0 0 0 100 100 0 0           %D1
     0 0 0 0 0 0 0 0 0 0 0 50 100 0            %D2
     0 0 0 0 0 0 0 0 0 0 0 150 150 0           %D3
     0 0 0 0 0 0 0 0 0 0 0 0 0 200             %E1
     0 0 0 0 0 0 0 0 0 0 0 0 0 150             %E2
     0 0 0 0 0 0 0 0 0 0 0 0 0 100             %E3
     0 0 0 0 0 0 0 0 0 0 0 0 0 0];            %F
```

```
xi=1;
for b=2:3
    for c=4:7
        for d=8:10
            for e=11:13
                if N(1,b)>0 && N(b,c)>0 && N(c,d)>0 && N(d,e)>0
                    luxian(xi,:)=[1,b,c,d,e,14];
                    changdu(xi)=N(1,b)+N(b,c)+N(c,d)+N(d,e)+N(e,14);
                    xi=xi+1;
                end
            end
        end
    end
end

for x1=1:24
    y=1;
    for x2=1:14
        if luxian(x1,y)==x2
            pass(x1,x2)=1;
            y=y+1;
        else pass(x1,x2)=0;
        end
    end
end
```

3、lingo 求解线性规划程序：

```
sets:
luxian/1..24/:xiang,lucheng;
zhandian/1..14/:limit,fei;
link(luxian,zhandian):x,z;
endsets
data:
limit=0 58 39 80 54 14 71 82 63 54 23 72 69 72;
fei=0 250 150 300 400 500 450 550 650 650 650 650 750 800;
x=@OLE('C:\Users\hlh\Desktop\question4.xls',x);
lucheng=900 850 900 900 850 800 850 850 900 900 1100 1050 1000 950 1000 1000 850 850 1050 1000 1050 1050 950 900;
enddata
max=2*@sum(zhandian(j):fei(j)*@sum(luxian(i):z(i,j)))-@sum(luxian(i):@floor(xiang(i)/40)*15000)-
15000*@sum(luxian(i):@if(40*@floor((xiang(i)/40))#EQ#xiang(i),0,1))-@sum(luxian(i):xiang(i)*lucheng(i)*1);!第一部分是每条线路判断运行不运行算列车固定成本和;
@for(zhandian(j):@sum(luxian(i):z(i,j))<limit(j));!让每一个点的运输量不超过需求;
@for(zhandian(j):@for(luxian(i):z(i,j)=@if(0#EQ#x(i,j),0,z(i,j))));!路线不通过的站点运输量为零;
@for(luxian(i):@sum(zhandian(j):z(i,j))<3*xiang(i));!表示每趟路线箱子的数量;
```

```
@for(luxian(i):@sum(zhandian(j):z(i,j))>3*(xiang(i)-1));!表示每趟路线箱子的数量;
@for(luxian(i):@for(zhandian(j):@gin(z(i,j))));
@for(luxian(i):@gin(xiang(i)));
```

第五问：

1、floyd 算法 matlab 程序：

```
 function [D,path]=floyd(a)
 n=size(a,1);
 D=a;
 path=zeros(n,n);
 for i=1:n
     for j=1:n
         if D(i,j)~=inf
             path(i,j)=j;
         end
     end
 end
 for k=1:n
     for i=1:n
         for j=1:n
             if D(i,k)+D(k,j)<D(i,j)
                 D(i,j)=D(i,k)+D(k,j);
                 path(i,j)=path(i,k);
             end
         end
     end
 end
```

2、matlab 计算最短距离以及潜在需求矩阵程序：

```
clear;
M=[0 250 150 0 0 0 0 0 0 0 0 0 0 0
   250 0    0 50 150 300 0 0 0 0 0 0 0 0
   150 0 0 0 400 350 300 0 0 0 0 0 0 0
   0 50 0 0 0 0 0 300 400 0 0 0 0 0
   0 150 400 0 0 0 0 150 250 0 0 0 0 0
   0 300 350 0 0 0 0 0 150 250 0 0 0 0
   0 0 300 0 0 0 0 0 400 200 0 0 0 0
   0 0 0 300 150 0 0 0 0 0 100 100 0 0
   0 0 0 400 250 150 400 0 0 0 0 50 100 0
   0 0 0 0 0 150 200 0 0 0 0 150 150 0
   0 0 0 0 0 0 0 100 0 0 0 0 0 200
   0 0 0 0 0 0 0 100 50 150 0 0 0 150
   0 0 0 0 0 0 0 0 100 150 0 0 0 100
   0 0 0 0 0 0 0 0 0 0 200 150 100 0];
```

```
for a=1:14
    for b=1:14
        if a==b
            M(a,b)=0;
        elseif a>b
            M(a,b)=inf;
        elseif M(a,b)==0
            M(a,b)=inf;
        end
    end
end
[S,path]=floyd(M);
for a=1:14
    for b=1:14
        if S(a,b)==inf
            S(a,b)=0;
        end
    end
end
need=zeros(14);
start=[1 1 1 9 2 10 5 4 7 8 3 1 1 6 4 8 1 4 3 3 3];
final=[4 13 8 13 9 12 11 11 13 14 14 6 14 9 8 12 2 13 11 12 13];
numbe=[10 44 96 4 71 32 19 68 34 22 15 22 56 28 30 89 57 93 99 57 49];
for i=1:21
    need(start(i),final(i))=numbe(i);
end
```

3、lingo 解线性规划程序：

```
sets:
luxian/1..24/:line_max,long,go,chexiang;
zhandian/1..14/:;
link1(luxian,zhandian):pass,line_on,line_up,line,sumsum;
link2(zhandian,zhandian):need,S,load;
link3(zhandian,zhandian,luxian):load_l;
link4(luxian,zhandian,zhandian):x;
endsets
data:

long=900 850 900 900 850 800 850 850 900 900 1100 1050 1000 950 1000 1000 850 850 1050 1000 1050 1050 950 900;
need=@OLE('C:\Users\hlh\Desktop\mathmodel\need.xls',need);
pass=@OLE('C:\Users\hlh\Desktop\mathmodel\pass.xls',pass);
S=@OLE('C:\Users\hlh\Desktop\mathmodel\S.xls',S);
```

```
enddata

!目标函数;
max=-15000*@sum(luxian(l):go(l))-@sum(luxian(l):chexiang(l)*long(l))+@sum(link2(i,j):load(i,j)*S(i,j)*2);!f(go , chexiang , load);

@for(luxian(l):@for(zhandian(i):@for(zhandian(j):x(l,i,j)=load_l(i,j,l))));
@for(link3(i,j,l):load_l(i,j,l)*(1-pass(l,i)*pass(l,j))=0);
@for(link1(l,j):line_on(l,j)=@sum(zhandian(i):load_l(j,i,l)));!j 站点放上 l 线路的货物数量;
@for(link1(l,j):line_up(l,j)=@sum(zhandian(i):load_l(i,j,l)));!j 站点取下 l 线路的货物数量;
@for(link2(i,j):@sum(luxian(l):load_l(i,j,l))=load(i,j));!i 运往 j 的货物等于 i 通过所有路线运往 i 的货物;

@for(link1(l,j):@free(line(l,j)));
@for(link1(l,j):line(l,j)=line_on(l,j)-line_up(l,j));!l 路线 j 站点的净装货物量等于放上货物减取下货物;
@for(luxian(l):@sum(zhandian(j):line(l,j))=0);!每条路线的净装总量为零，即放上的货物等于取下的货物;

!j 站点所有路线的净装量之和等于 j 站点的运出之和减去运入之和;
@for(zhandian(j):@sum(luxian(l):line(l,j))=@sum(zhandian(i):load(j,i))-@sum(zhandian(i):load(i,j)));
@for(link2(i,j):load(i,j)<need(i,j));!i 运往 j 的货物数量不超过潜在运输量;

@for(link1(l,m):sumsum(l,m)=@sum(link1(l,j)|j#LE#m:line(l,j)));!sumsum 表示 l 路线上在 m 站点时车上的货物量;
@for(luxian(l):line_max(l)=@max(zhandian(j):sumsum(l,j)));!求出每条线路车上运输最多货物时的货物量;
@for(luxian(l):3*chexiang(l)>line_max(l));

@for(luxian(l):go(l)=@if(line_max(l)#EQ#0,0,1));!约束发车关系，线路上无运货，则不发车;
@for(link3(i,j,l):@gin(load_l(i,j,l)));!每条线路站点之间的运输货物量为整数;

@for(luxian(l):@gin(chexiang(l)));
```