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：2016 年 8 月 5 日

赛区评阅编号（由赛区组委会评阅前进行编号）：

2015 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）：

摘要

因为本文主要是求最佳装运和编组方案使得铁路部门获得最高收益，装运要求是运输货物数量最多，总重量最小。而收益要求运费获得最多，而火车固定成本和可变成本最小，以此建立模型求解货运列车的最佳运输方案。

问题 1：这里先列出所有货物装运的约束条件，特别是货物 B 在奇偶时不同的放置方式，以及货物占用车厢的长宽高不能车厢的自身的长宽高，货物总重量不超过车厢的承载量，Ⅱ车厢上下层的约束条件。通过约束条件，用 MATLAB 算出所有可行的装运方案见附录一，并给出最优方案（运输货物数量最多条件下，运输总重量最小）为运输数量 24 件，总重量 180.5 吨。

问题 2：这实际就是一个下料问题，我们首先列出 B、C、E 的约束条件，运用整数线性规划模型求解，用 MATLAB 算出只考虑 B、C、E 的可行性装运方式，用 Excel 对每种装运方式进行数据分析以整合排除明显劣解，确定该模型中的决策变量，从而得到最终的目标函数当 B、C、E 为 68,50,41 件时车厢数为 25，当 B、C、E 为 48、42、52 件时车厢数为 23。装运方式见附录二。

问题 3：我们首先用 EXCEL 将近 100 天上下午需要集装箱数据进行数据处理，通过 MATLAB 对两组数据进行分布拟合发现两组数据均服从正态分布，并算出其上下午接受概率分别为 0.1812 和 0.9288，利用概率密度模型，和正态分布的逆概率分布求解可得上午和下午集装箱和车厢分别为 121 件、41 节和 100 个、34 节。

问题 4：要利用现有的交通运输条件制定一套调运方案，使得预期的运费收入最大。我们首先用 Dijkstra 算法算出 A 点到各个点的最短距离，以此建立编组效率最高（路径最短）受益最大（每列车集装箱数尽可能多）的收益模型，再通过线性规划，Dijkstra 算法计算出最大和次大获益路径，并得到了最佳编组方案 A-B1-C2-D2-E3-F,共 3 列，车厢依次为 40、40、29;A-B2-C2-D1-E1-F 共 1 列，车厢数 40；A-B2-C4-D3-E3-F,1 列，车厢为 40；A-B1-C1-D1-E2-F,共 1 列，车厢数 40；A-B2-C3-D2-E2-F 共 1 列，车厢数 40。总利润为 972000 元。

问题 5：经过分析从 A 点到 F 点一共有 24 条路线，与问题四不同的是各个站点相互之间又有需求，尽可能满足大多数站对集装箱的需求使得收益最大。我们先用 Dijkstra 算法算出各个站点之间的最短路径，运用改进的线性规划模型，得出路线 A-B1-C2-D2-E3-F,1 列，40 节车厢；A-B2-C4-D3-E3-F,1 列，40 节车厢；A-B2-C2-D1-E1-F，2 列，车厢依次为 40 节，37 节。总利润为 430550 元。

关键词：整数型线性规划　EXCEL 数据处理　　概率密度函数模型　　Dijkstra 算法

一、 问题重述

货运列车编组调度的科学性和合理性直接影响着货物运输的效率。在不同的问题设定下，进而分析得出货运列车编组不同背景下的最佳编组方案，我们需要解决的问题如下：

1.1 问题一

1） 甲地到乙地每天有 5 种类型的货物需要运输，每种类型货物包装箱参数确定（见附录一表 1)。
2） 每天有一列货运列车从甲地发往乙地，该列车由 1 节 Ⅰ型车厢（单层）和 2 节（双层）Ⅱ型车厢编组（具体规格见附录一表 2)。
3） 货物在车厢中必须按占用车厢长度最小方式放置且不允许货物重叠放置；Ⅱ型车厢下层装载货物后剩余长度小于等于 0.2 米，才能在上层放置货物。

◣ 试设计运输货物数量最多的条件下，运输总重量最小的装运方案。

1.2 问题二

1） 编组中Ⅰ型车厢的数量多于Ⅱ型车厢数量。
2） Ⅱ型箱式车厢下层装载货物后剩余长度小于等于 5 米，才能在上层放置货物。
3） 货物装车其它规则同问题 1。

◣ 如果现有 B,C,E 三种类型的货物各 68、50、41 件，试设计一个使用车厢数量最少的编组方案将货物运输完毕。

◣若 B,C,E 三种类型的货物各有 48,42,52 件，请重新编组。

1.3 问题三

1） 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。
2） 每列火车开行的固定成本为 30000 元，每加挂一节车厢的可变成本为 1500 元。
3） 铁路部门拟将货物放置到长、宽、高分别为 4 米，3 米及 1.99 米的集装箱中运输，每个集装箱的总重量不超过 18 吨，集装箱的运费为 1000 元/个。
4） 每天需要运输的集装箱数量是随机的（附录一表 3 给出了过去最近 100 天上午和下午分别需要运输的集装箱的数量）。
5） 上午的需求如果不能由上午开行列车运输，铁路部门要支付 50 元/个的库存费用；下午列车开行后如果还有剩余集装箱，铁路部门将支付 200 元/个的赔偿，转而利用其它运输方式运输。

◣ 试制定两列火车的最佳编组方案。

1.4 问题四

1） 每天铁路部门将以 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，全部用Ⅰ型车厢编组，每列火车最大编组量为 40 节车厢。
2） 每列火车列车开行的固定成本为 15000 元，每节车厢开行的可变成本为 1 元/公里，每个集装箱的运费为 2 元/公里（集装箱的运费按两个车站之间的最短铁路距离计费）。
3） 某铁路网线情况见附录一表 4
4） 从车站 A 到其它站点的潜在集装箱运输需求量见附录一表 5，集装箱规格同第 3 问（铁路部门没有义务把集装箱全部运输完毕）。
◣ 请为铁路部门设计一个编组运输方案。

1.5 问题五

1.6

1）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。
2）铁路网线及费用设定同问题 4。
3）每天各个车站之间潜在的集装箱运输量见附录一表 6。
◣ 请为铁路部门设计一个编组运输方案。

二、 问题分析

2.1 问题一

对于问题一，要求货物的最佳装运，使得铁路部门获得最高收益，装运的目标函数是运输货物数量最多，总重量最小。这里先列出所有货物装运的约束条件，特别是货物 B 在奇偶时不同的放置方式，以及货物占用车厢的长宽高不能车厢的自身的长宽高，货物总重量不超过车厢的承载量，Ⅱ车厢上下层的约束条件。通过约束条件，用 MATLAB 算出所有可行的装运方案并给出最优方案（运输货物数量最多条件下，运输总重量最小）

2.2 问题二

与问题一类似，但只考虑货物 B、C、E 的需求量。这实际就是一个下料问题，我们首先列出 B、C、E 的约束条件，运用整数线性规划模型求解，用 MATLAB 算出只考虑 B、C、E 的可行性装运方式，用 Excel 对每种装运方式进行数据分析以整合排除明显劣解，确定该模型中的决策变量，从而得到最终的目标函数。

2.3 问题三

我们首先用 EXCEL 将近 100 天上下午需要集装箱数据进行数据处理，通过 MATLAB 对两组数据进行分布拟合发现两组数据均服从正态分布，并算出其上下午接受概率，利用概率密度模型，和正态分布的逆概率分布求解。

2.4 问题四

要利用现有的交通运输条件制定一套调运方案，使得预期的运费收入最大。我们首先用 Dijkstra 算法算出 A 点到各个点的最短距离，以此建立编组效率最高（路径最短）受益最大（每列车集装箱数尽可能多）的收益模型，再通过线性规划，Dijkstra 算法计算出最大和次大获益路径，并得到了最佳编组方案

2.5 问题五

与问题四类似，只是改变了条件，换成了各个站点之间的相互需求，尽可能满足大多数站对集装箱的需求使得收益最大。我们先用 Dijkstra 算法算出各个站点之间的最短路径，运用改进的线性规划模型，得出路线。

三、符号说明

$\boldsymbol{i}$	i =1,2,3,4,5 分别对应代表货物 A，B，C，D，E
$\boldsymbol{n_i}$	i型货物的数量
$\boldsymbol{l_i}$	i型货物所占车厢长度
$\boldsymbol{w_i}$	i型货物总重量
$\boldsymbol{L_1, L_2}$	Ⅰ,Ⅱ型车厢总长度
$\boldsymbol{W_1, W_2}$	Ⅰ,Ⅱ型车厢总重量
$\boldsymbol{a_j}$	只考虑 B,C,E 装运的Ⅰ型车厢第 j 种装运方式的使用次数
$\boldsymbol{b_j}$	只考虑 B,C,E 装运的Ⅱ型车厢第 j 种装运方式的使用次数
$\boldsymbol{G_1, G_2}$	铁路部门上午、下午的利润
$\boldsymbol{r_1, r_2}$	上午、下午需要运送的集装箱数量
$\boldsymbol{x_1, x_2}$	上午、下午铁路部门发出的车厢数

四、问题假设

1. 假设每次运输均由相同规格货运列车完成
2. 假设集装箱在运输过程中没有任何损耗，且不会因交通问题影响集装箱的需求
3. 假设只考虑所涉及的货运列车的运费，车厢的可变成本，和发出的固定成本，其他费用不计。
4. 假设没有运到站点的集装箱，铁路部门不收取运费

5.假设 A 行驶到 F 的过程中，各个站点的装卸过程中，车厢的每个位置不重复使用。

五、模型的建立与求解

5.1 问题一

5.1.1 模型的准备

为简化问题及更好地对货物装车进行不同条件下的约束，我们可以对Ⅰ型和Ⅱ型进行分别的可行货物装载方式分析，再将这些方案进行组合，最终得出最优方案(运输货物数量最多条件下，运输总重量最小)。

1.基本思路

将三维货物简化为二维矩形

↓

列出所有货物装运的约束条件

↓

通过约束条件，用 MATLAB 算出所有可行的装运方案

↓

计算每种方案相应的运输数量和运输总重量

↓

通过题目给出的条件，用 Excel 筛选出最优装运方案

5.1.2 模型的建立

1. 将三维货物简化为二维矩形

由于不能重叠放置，所以我们将车厢中的货物放置问题简化为二维矩形进行排列解决。

2.在考虑单个车厢时，需考虑以下约束条件：

1) 货物数目不能超过题目给定数量，即：

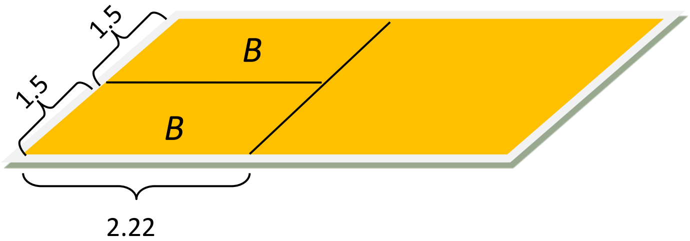
$$\begin{cases} \boldsymbol{n_1} \le 7 \\ \boldsymbol{n_2} \le 6 \\ \boldsymbol{n_3} \le 5 \\ \boldsymbol{n_4} \le 7 \\ \boldsymbol{n_5} \le 6 \end{cases}$$

注：由于我们已将Ⅰ型车厢和Ⅱ型车厢分开考虑，无法考虑这一约束条件，所以这里我们仅列出这一约束条件，在最后选择最优方案再判断方案是否符合这一约束条件，如不符合，则选择次优方案，依次类推，直到选出符合这一条件的最优方案为止。

2）货物按照占用车厢最小长度的方式放置

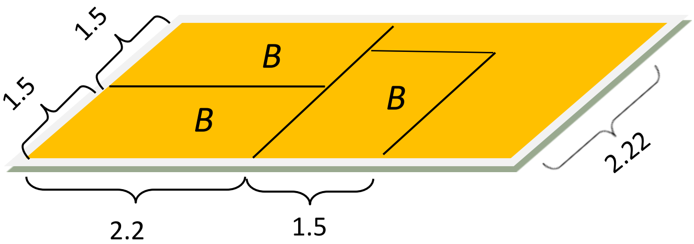
因为货物 A,C,D,E 的宽度与两种车厢的宽度一样，所以货物 A,C,D,E 的占用车厢长度就是它们自身长度一样，只有货物 B 的宽度与车厢宽度不同，所以只用对货物 B 在车厢内的放置方式进行讨论：

① 当车厢内放置货物 B 的数量为偶数个时（即$\boldsymbol{n_2}$=0，2，4，6 时），货物 B 在车厢内放置方式为两两并排放置时（如图 1），货物 B 占用车厢长度最小



图一 偶数个货物 B 在车厢内的放置方式

② 当车厢内放置货物 B 的数量为奇数个时（即$\boldsymbol{n_2}$=1，3，5 时），货物 B 在两两并排放置的基础上，将剩余的一件 B 货物横向放置在车厢内，可使货物 B 占用车厢长度最小



图二 奇数个货物 B 在车厢内的放置方式

即

$$l_2=\begin{cases}1.11(\boldsymbol{n_2}-1)+1.5 & (\boldsymbol{n_2}=1,3,5)\\ 1.11\boldsymbol{n_2} & (\boldsymbol{n_2}=0,2,4,6)\end{cases}$$

3）货物占用车厢高度不能超过车厢自身高度

因前面已假设过货物不能竖直放置（因车身高度问题，所有类型货物均无法竖直放置），所以此时我们只用考虑货物实际高度与车厢高度之间的关系，可以得出结论：Ⅰ型车厢以及Ⅱ型车厢下层能放所有类型货物，Ⅱ型车厢上层只能放 C,D,E 三种货物。

4）货物占用车厢高度不能超过车厢自身长度

$$\boldsymbol{l_1+l_2+l_3+l_4+l_5\le L}$$

5）货物占用车厢宽度不能超过车厢自身宽度

在 2）中我们已经提到，货物 A,C,D,E 的宽度与两种车厢的宽度一样，只有货物 B 的宽度与车厢宽度不同，且按照 3）中所提到的货物放置方式，所以这一条件不必单独考虑。

6）车厢承载货物总重量不能超过车厢允许最大载重量

$$w_1 + w_2 + w_3 + w_4 + w_5 \leq W$$

7）Ⅱ型车厢下层装在货物后剩余长度小于等于 0.2 米时，才能在上层放置货物

3. 通过约束条件，用 MATLAB 算出所有可行的装运方案

考虑 5.1.2 中 2.2),3),4),5),6),7)条件，用 MATLAB 编程，分别列举出Ⅰ型车厢和Ⅱ型车厢所有满足条件的可行的货物装载方式（程序分别见附录二表 1 和表 3），Ⅰ型车厢和Ⅱ型车厢所有满足条件的可行的货物装载方式分别见附录二表 2 和表 4。

5.1.3 模型的求解

通过 Excel 算出所有Ⅰ型车厢和Ⅱ型车厢所有满足条件的可行的货物装载方式中货物数量及货物所占重量。再用 Excel 进行筛选，分别选出Ⅰ型车厢和Ⅱ型车厢装运货物数量最多的若干种方案，再分别对其按照货物所占重量进行排序，得到两种车厢中装运货物数量最多条件下货物所占重量最小的装运方案，分别找到两种车厢中的最优解，然后将型车厢和Ⅱ型车厢中的最优解进行相加（此时需满足 5.1.2 中 2.1）中条件，并按其中提到的方式进行最优方案组合），最终得到满足题目条件的货物装运方案。以下为最优装运方案：

运输重量		180.5	
运输数量		24	
Ⅰ型车厢	A	2	2
	B	0	0
	C	4	4
	D	0	0
	E	0	0
Ⅱ型车厢1	A下	3	3
	B下	1	1
	C下	0	0
	D下	0	0
	E下	2	2
	C上	0	0
	D上	1	1
	E上	2	2
Ⅱ型车厢2	A下	2	2
	B下	1	1
	C下	0	0
	D下	1	3
	E下	2	0
	C上	0	0
	D上	3	1
	E上	0	2

5.2 问题二

5.2.1 模型的准备

如 5.1.1 中提到的，为简化问题及更好地对货物装车进行不同条件下的约束，我们可以对Ⅰ型和Ⅱ型进行分别的可行货物装载方式分析，对Ⅰ型和Ⅱ型进行分别的可行货物装载方式分析这一方法在只考虑在问题二中减少了货物类型的情况下同样适用。

1. 基本思路

确定这是一个需用整数线性规划模型解决的下料问题

↓

确定两种车厢只考虑 B,C,E 时的约束条件

↓

通过约束条件，用 MATLAB 算出每种只考虑 B,C,E 可行的装运方式

↓

用 MATLAB 对每种装运方式进行数据分析及整合排除明显劣解

↓

确定整数线性规划模型中的决策变量

↓

确定目标函数及其约束条件从而得到最终目标函数

5.2.2 模型的建立

1. 基本思路

此问题在实质上是一个下料问题，所以首先要考虑到所有满足条件的下料方式，即所有每一车厢只考虑装运 B,C,E 货物的可行装运方式。以每种装运方式使用的次数作为决策变量，使用次数的总和最小则代表使用车厢数最少，从而可以确定目标函数，再找出函数中的约束条件，即可得到最终的目标函数。

2. 确定两种车厢只考虑 B,C,E 时的约束条件

1）货物按照占用车厢最小长度的方式放置

$$\begin{cases} 1.11(\boldsymbol{n_2}-1)+1.5 \quad (\boldsymbol{n_2}=1,3,5) \\ 1.11\boldsymbol{n_2} \quad (\boldsymbol{n_2}=0,2,4,6) \end{cases}$$

2）货物占用车厢高度不能超过车厢自身高度，即Ⅱ型车厢上层不能放 B 型货物

4）货物占用车厢高度不能超过车厢自身长度

$$\boldsymbol{l_2 + l_3 + l_5 \le L}$$

5）车厢承载货物总重量不能超过车厢允许最大载重量

$$\boldsymbol{w_2 + w_3 + w_5 \le W}$$

6）Ⅱ型车厢下层装在货物后剩余长度小于等于 5 米时，才可以在上层放置货物

3. 通过约束条件，用 MATLAB 算出每种只考虑 B,C,E 可行的装运方式，并筛选。

Ⅰ型车厢货物可行装载方案程序见附录三表 1，Ⅱ型车厢货物可行装载方案附录三表 3。Ⅰ型车厢货物可行装载方案（筛选后）见附录三表 2，Ⅱ型车厢货物可行装载方案（筛选后）见附录三表 4。

4. 确定整数线性规划模型中的决策变量

用$\boldsymbol{a_j}$表示只考虑 B,C,E 装运的Ⅰ型车厢第 j 种装运方式的使用次数，用$\boldsymbol{b_j}$表示只考虑 B,C,E 装运的Ⅱ型车厢第 j 种装运方式的使用次数,所以$\boldsymbol{a_j}$（j=1, 2, ⋯, 22）、$\boldsymbol{b_j}$（j=1, 2, ⋯, 125）为此模型的决策变量。

5. 确定目标函数

因为问题二要求使用车厢数最少，即我们保证各装运方式使用次数之和最小，所以目标函数为：

$$\min\ \sum_{j=1}^{22} a_j + \sum_{j=1}^{125} b_j$$

6.确定目标函数的约束条件

1）Ⅰ型车厢数量多于Ⅱ型车厢数量，即：

$$\sum_{j=1}^{22} a_j > \sum_{j=1}^{125} b_j$$

2）货物必须运输完毕，即装运方案可运走的 i 型货物数量不小于现有的数量

$$\sum_{j=1}^{22} a_j \cdot T_i + \sum_{j=1}^{125} b_j \cdot T_i \le m_i (i = 2,3,5)$$

T_2、T_3、T_5分别为各方案中 B,C,E 类货物的运载量

3）决策变量为非负整数，即：

$$a_j \geq 0, \ b_j \geq 0, \mathrm{int}$$

7.得到目标函数

$$\mathrm{s.t.}\begin{cases} \min \sum_{j=1}^{22} a_j + \sum_{j=1}^{125} b_j \\ \sum_{j=1}^{22} a_j > \sum_{j=1}^{22} b_j \\ \sum_{j=1}^{22} a_j \cdot T_i + \sum_{j=1}^{125} b_j \cdot T_i \leq m_i (i=2,3,5) \\ a_j \geq 0, \ b_j \geq 0, \mathrm{int} \end{cases}$$

5.2.3 模型的求解

使用 MATLAB 程序对所得到的整数线性规划模型进行求解（程序详见附录三表 5），得到以下解

1. B,C,E 三种类型的货物各 68、50、41 件时

车厢型号	Ⅰ型车厢（13列）				Ⅱ型车厢（12列）	
车厢列数	2	8	2	1	5	7
B	0	2	3	5	3	4
C	6	2	2	0	0	3
E	0	2	0	0	4	0
C上					0	0
E上					1	0

2. B,C,E 三种类型的货物各有 48,42,52 件时

车厢型号	Ⅰ型车厢（13列）			Ⅱ型车厢（10列）					
车厢列数	3	2	6	1	1	4	1	1	2
B	0	1	2	0	3	3	4	5	5
C	6	4	2	0	0	0	3	0	0
E	0	1	2	4	3	4	0	1	1
C上				0	0	0	0	0	1
E上				5	2	1	0	1	0

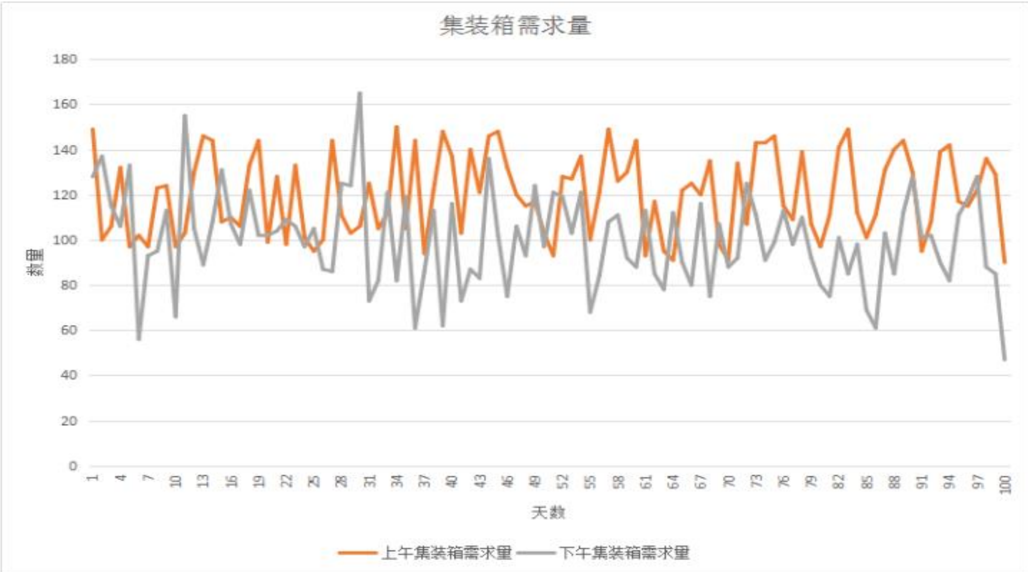
5.3 问题三

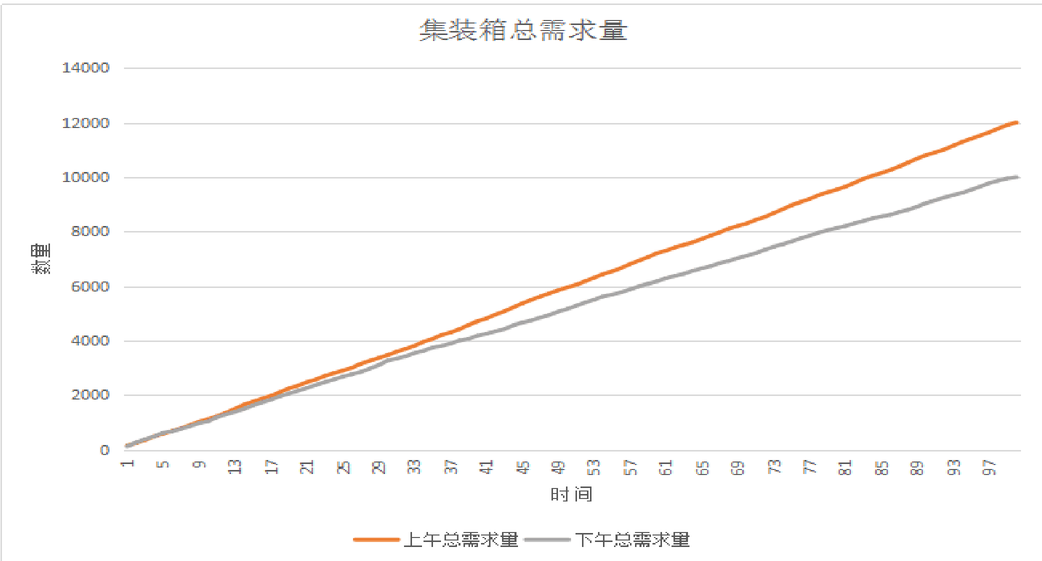
5.3.1 模型的准备

为了使公司获取最大利益，要求利用现有的一百天的数据，对未来集装箱的总量进行预测。根据题意，每天集装箱的数量是随机的，而且需要预测的数据是

未来短期内的，可以根据天数和集装箱总量之间的发展趋势通过数据分析，从而建立合适的预测模型。

1. 对题目给出的最近 100 天上下午需要的集装箱数据进行数据分析





通过 Excel 作出集装箱每日需求量的折线图以及集装箱总需求量的图，并没有发现明显规律

2. 通过 MATLAB 对两组的数据进行分布拟合

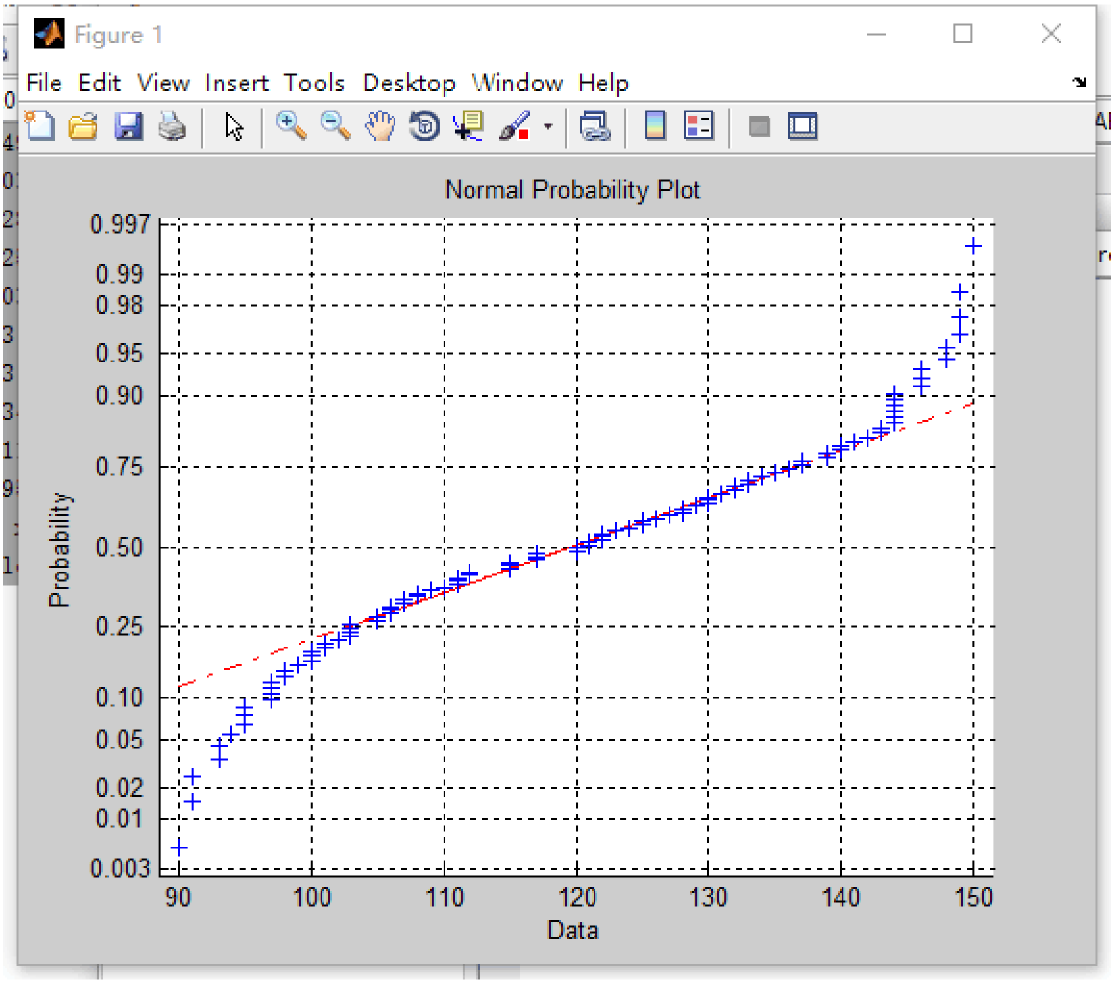
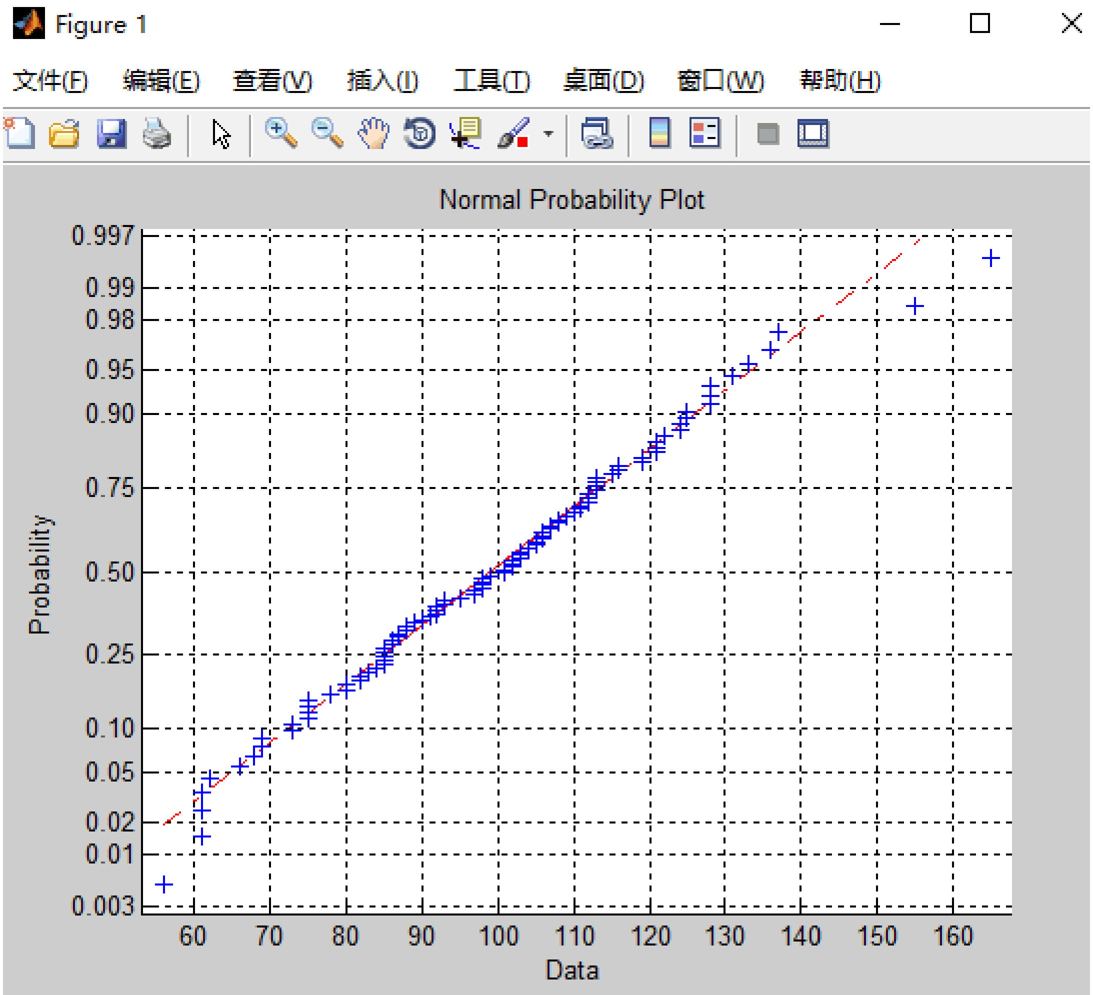


Figure 1
File Edit View Insert Tools Desktop Window Help
Normal Probability Plot
Probability
0.997
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
0.003
90
100
110
120
130
140
150
Data

上午



下午

通过 MATLAB 对这组数据的拟合，发现两组数据均不同程度地服从正态分布，拟合度分别为 0.1812、0.9288；上午对集装箱需求的均值为 120.01，下午对集装箱需求的均值为 0.9288。

3.基本思路

对题目给出的最近 100 天上下午需要的集装箱数据进行数据分析

↓

通过 MATLAB 对两组的数据进行分布拟合发现两组数据均服从正态分布

↓

确定关于铁路部门货运利润的目标函数

↓

建立求目标函数的概率密度模型

5.3.2 模型的建立

1.确定目标函数

因为上午和下午需要运输的集装箱数量都是随机的，所以我们可以对上午、下午进行分别考虑，即目标函数可表示为

$$\max G = G(x)_1 + G(x)_2$$

2.建立求目标函数的概率密度模型

众所周知，应该根据需求量来确定集装箱个数。然而每天需要运输的集装箱是随机的，假定铁路局已经通过自己的经验或其他的渠道掌握了需求量的随机规律，即在他的销售范围内每天需要的集装箱的个数是 r 个的概率是 $f(r)(r=0,1,2,\ldots)$.有了 $f(r)$ 就可以建立关于购进量的优化模型了。

假设有 x 节车厢，则应该有 3x 个集装箱。因为需求量 r 是随机的，r 可以小于 3x、等于 3x、或大于 3x、致使铁路局每天的收入也是随机的，所以作为优化模型的目标函数不能是铁路局每天的收入，而应该是长期货运的日平均收入。从概率论的观点看，这相当于铁路局每天收入的期望值，以下简称平均收入。

记铁路局每天的平均收入为 $G(x)$，如果这天的需求量 $r \le 3x$，则运送 r 件，退回 3x−r 件；如果这天的需求量 $r > 3x$，则 x 份将全部售出。考虑到需求量为 r 的概率是 f(r)，所以

1)在上午时

$$G(x)_1 = \begin{cases} 1000 r_1 - 1500x - 30000 & r_1 \le 3x \\ 1000 \times 3x - 1500x - 30000 - 50 \times (r_1 - 3x) & r_1 \succ 3x \end{cases} \quad (1)$$

通常情况下 r 的取值和集装箱个数 3x 都相当大，将 r 视为连续变量更便于分析和计算，这时概率 $f(r)$ 转化为概率密度函数 $p(r)$，（1）式变成

$$G(x)_1 = \int_0^{3x} (1000 r_1 - 1500x - 30000) dr_1 + \int_{3x}^{\infty} (1650x - 50 r_1 - 30000) p(r_1) dr_1 \quad (2)$$

计算

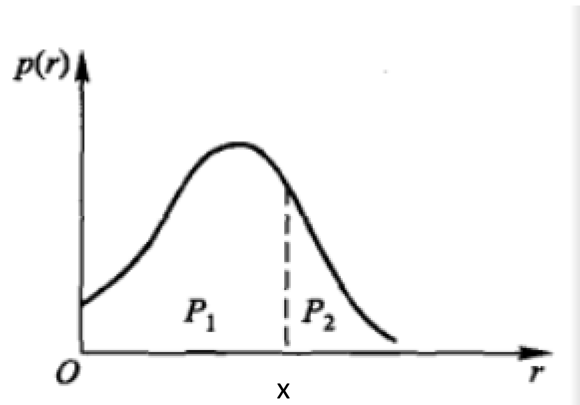
$$\frac{dG}{dx} = -\int_0^{3x} 1500\, dr_1 + \int_{3x}^{\infty} 1650\, p(r_1) dr_1$$

令 $\frac{dG}{dx} = 0$ 得到

$$\frac{\int_0^{3x} p(r_1) dr_1}{\int_{3x}^{\infty} p(r_1) dr_1} = \frac{11}{10} \quad (3)$$

又因为$\int_0^{\infty} p(r)dr = 1$，所以$\int_0^{3n} p(r_1)dr_1 = \frac{11}{21}$

因为在有 x 节车厢即有 3x 个集装箱时，$P_1 = \int_0^{n} p(r)dr$ 是需求量 r 不超过 x 的概率，即上午能运完的情况下；$P_2 = \int_n^{\infty} p(r)dr$ 是需求量 r 超过 x 的概率，即上午的需求如果不能由上午开行列车运输，需要留到下午运输的。由（3）式表明，集装箱的个数应该是在不足和多余的比恰好等于$\frac{11}{10}$。



由 p(r)确定 x 的图解法

2）在下午时

$$G(\mathrm{x})_2 = \begin{cases} 1000 r_2 - 1500\mathrm{x} - 30000 & r_2 \le 3x \\ 1000 \times 3x - 1500x - 30000 - 200(r_2 - 3x) & r_2 \succ 3x \end{cases}$$

同理可证$\int_0^{3\mathrm{x}} p(r_2)dr_2 = \frac{7}{12}$

5.3.3 模型的求解

通过 MATLAB 程序解得x_1, x_2分别为 40.364、37.382，即：
最佳编组方案为上午发 41 节 I 型车厢，下午发 38 节 I 型车厢

5.4 问题四

5.4.1 模型的准备

货运列车编组运输问题研究了如何利用现有的交通运输条件制定出一套调运方案,使得预期的运费收入最大。我们介绍了什么是最短路径问题,解决最短路问题的基本思路,*Dijkstra* 算法,最后提出了货运列车编组运输问题,基于博弈论思想,结合以上分析,以编组效率最高(路径最短),收益最大(每一列火车上尽可能多的运上所需要的集装箱)为优化目标,建立货车编组调度多目标优化模型。

基本思路:

将题目类比为图论中的*Dijkstra*算法

↓

通过*Dijkstra*算法算出 A 点到各点最短距离

↓

根据获益与距离关系列出收益函数

↓

通过*Dijkstra*算法计算出最大及次大获益路径

↓

在货物需求量的约束条件下求得编组方案

5.4.2 模型的建立

1.将题目类比为图论中的*Dijkstra*算法

Dijkstra 是求解最短路问题的有效算法之一,它的基本思路是逐点求最短路。标号法是通过对图上各点进行标号来寻求最短的方法。每个点的标号共分两种:一种是临时标号,用 T 表示;一种是永久标号,用 P 表示。T 标号表示从始点到该点最短路的上界,根据到该点路线的不同它有可能变化。P 标号表示从始点到该点的最短路权,它的值不再改变。标号过程分两步:

第一步,修改 T 标号。假定 v_i 是新产生的 P 标点号,考察以 v_i 为始点的所有弧段 v_iv_j。如果 v_j 是 P 标号法,则对 v_j 点不再进行标号;如果 v_j 点是 T 标号点,则进行如下的修改

$$T(v_j)=\min[T(v_j),P(v_i)+w_{ij}]$$

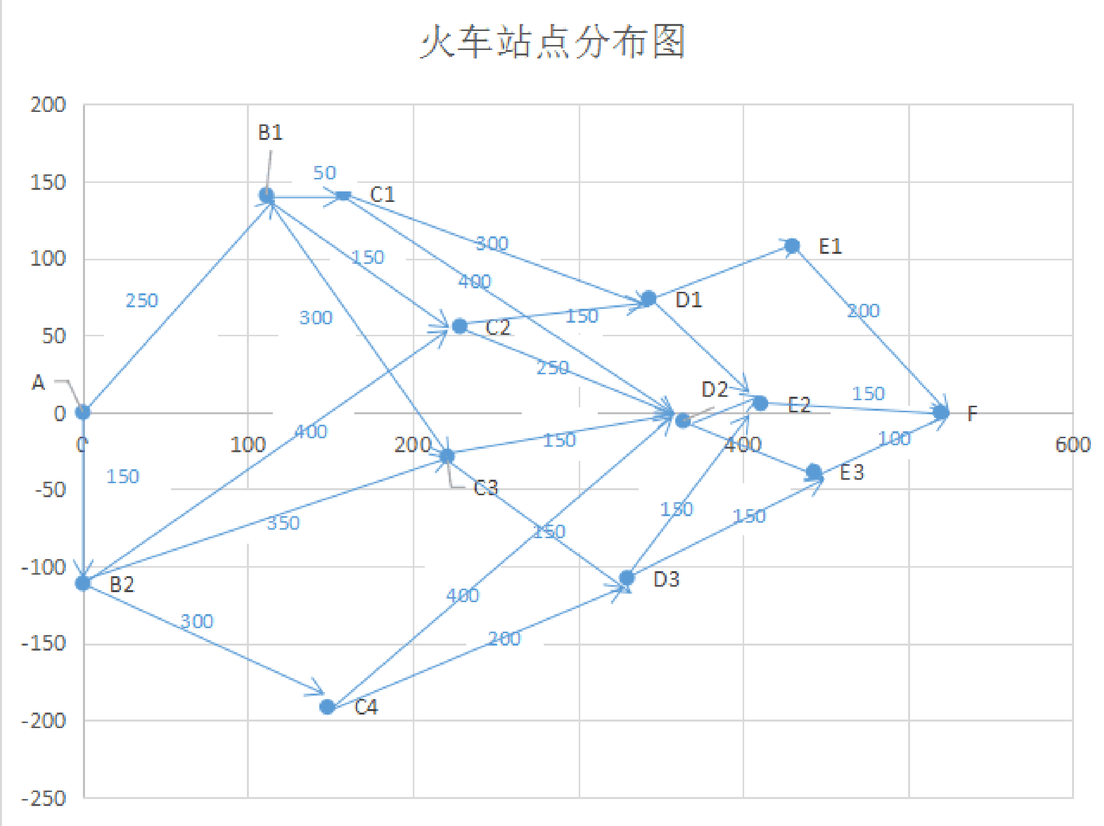
其中,方括号内的 $T(v_j)$ 代表 v_j 点旧的 T 标号值。

第二步,产生新的 P 标号点,其原则如下:在现有的 T 标号中将值最小者改为 P 标号。

重复以上步骤直到终点的 T 标号改为 P 标号为止。

2.通过算法算出 A 点到各点最短距离

该铁路各个站点,道路情况,各段距离的具体位置如图。



则各点的最短距离如图所示:

Variable Editor - Muti_Cost

Stack: Base

Muti_Cost <14x14 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	250	150	300	400	500	450	550	650	650	650	650	750	800
2	250	0	400	50	150	300	650	300	400	450	400	400	500	550
3	150	400	0	450	400	350	300	550	500	500	650	550	600	700
4	300	50	450	0	200	350	700	300	400	500	400	400	500	550
5	400	150	400	200	0	400	600	150	250	400	250	250	350	400
6	500	300	350	350	400	0	350	300	150	150	400	200	250	350
7	450	650	300	700	600	350	0	450	400	200	550	350	350	450
8	550	300	550	300	150	300	450	0	150	250	100	100	250	250
9	650	400	500	400	250	150	400	150	0	200	250	50	100	200
10	650	450	500	500	400	150	200	250	200	0	350	150	150	250
11	650	400	650	400	250	400	550	100	250	350	0	200	300	200
12	650	400	550	400	250	200	350	100	50	150	200	0	150	150
13	750	500	600	500	350	250	350	250	100	150	300	150	0	100
14	800	550	700	550	400	350	450	250	200	250	200	150	100	0

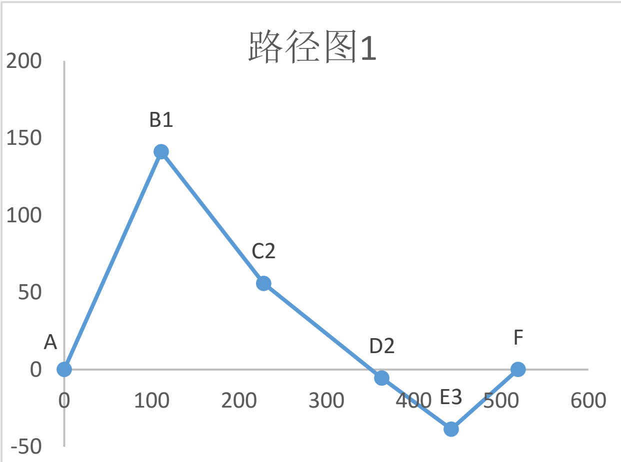
3.根据获益与距离关系列出收益函数

函数需满足如下假设:
(1)只考虑运输费用,不考虑装卸等其他费用。
(2)假设运输的集装箱路途中没有损耗。
(3)假设各站点需要集装箱同质。
编组效率评价指标建立:
对于列车编组的实际情况,我们小组进行了实地考察。发现列车编组绝大部分是在铁路编组站进行,而编组规定主要包括两个方面:
1)编入列车的车辆必须符合列车的性质。
2)编入列车的车辆要符合站顺或地段顺序。
经济因素主要包括:
1)车辆在编组站停留的时间消耗;
2)车辆走行里程运输成本;
3)列车的长度和重量等方面。
4)据分析,可得这五个衡量编组调度效率的主要指标。
通过计算可知

编号	站点	A到其最短距离	运费/个	需求/个
2	B1	250	500	58
3	B2	150	300	39
4	C1	300	600	80
5	C2	400	800	54
6	C3	500	1000	14
7	C4	450	900	71
8	D1	550	1100	82
9	D2	650	1300	63
10	D3	650	1300	54
11	E1	650	1300	23
12	E2	650	1300	72
13	E3	750	1500	69
14	F	800	1600	72

5.4.3 模型的求解

通过 Dijkstra 算法,我们求出了收益最大的第一条路径如下图所示。



路径为 A——B1——C2——D2——E3——F

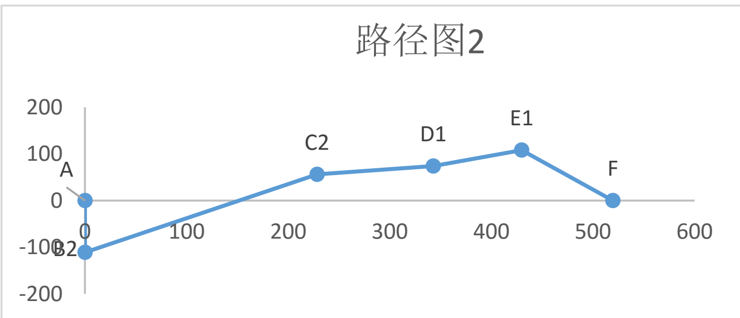
此为第一次迭代的结果,然后再考察以点 A 为始点 F 为终点的弧,按照第一次迭代的方法,迭代三次,我们的总目标是收益最大,其中收入的主要来源是运费,则对于各个站点的需求量我们按下图的排序模式满足需求,当第三列火车发出时只需要开到 D2 就行,满足最后一列车的需求 C2 为 54 件,D2 为 32 件。盈利 Z=P-W
。

P=54×400×2+32×650×2=84800

成本为 W=15000+(86/3+1)×800=38200 元 综上所述:Z=46600,第三列盈利为 46600 元。

	A	B	C	D
1	各个站点	距离	运费收入	优先满足
2	B1	250	29000	1
3	C2	400	43200	5
4	D2	650	81900	4
5	E3	750	103500	3
6	F	850	115200	2

第二条路线是



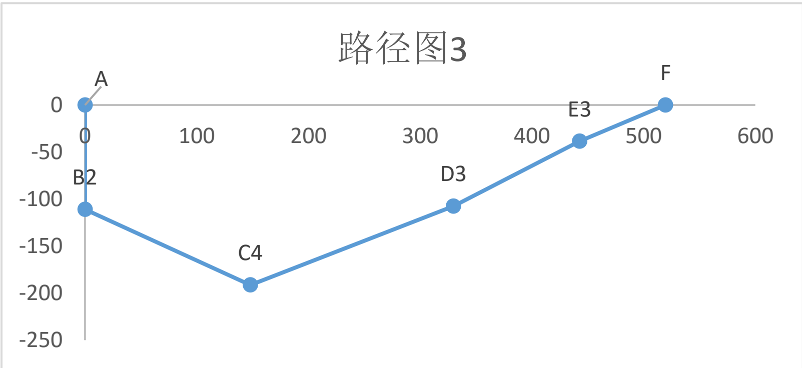
路径为 A——B2——C2——D1——E1——F

如上图所示,迭代一次(其中 C2 无货)即可。在此路径中 B2 为 15 件货,D1 为 82 件货,E1 为 23 件货。盈利为:

15×150×2+82×700×2+23×800×2-15000-(15+82+23)/3×800=109100 元

各个站点	距离	运费收入	优先满足
B2	150	11700	3
C2	550	0	
D1	700	90200	1
E1	800	29900	2
F	1000	0	

第三条路径是

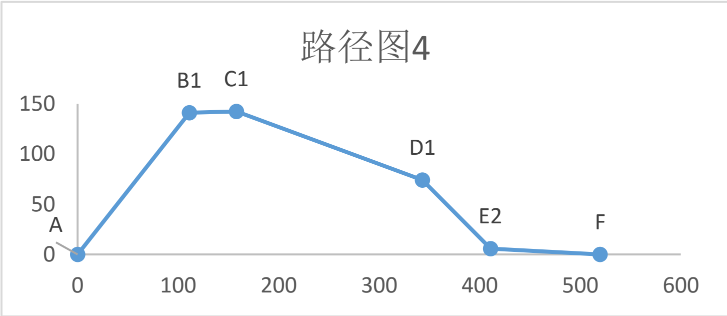


路径为 A——B2——C4——D3——E3——F

各个站点	距离	集装箱数	优先满足	运费收入
B2	150	24	3	7200
C4	450	71	2	63900
D3	650	54	1	70200
E3	800	0		0
F	900	0		0

同上只迭代一次,D3 为 54 件,C4 为 66 件。考虑到火车的固定成本问题,就不进行第二趟列车了。Z=P-W=123000-26000-15000=82000 元

第四条路径是



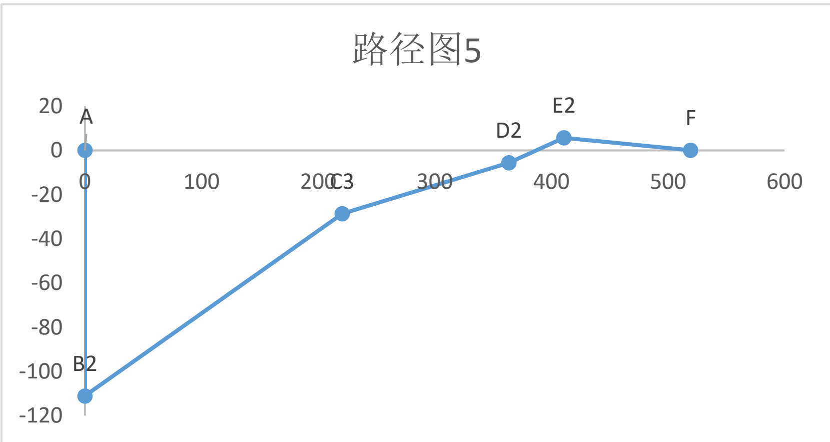
路径为 A——B1——C1——D1——E2——F

各个站点	距离	集装箱数	优先满足	运费收入
B1	250	0		
C1	300	80	1	48000
D1	550	0		
E2	650	72	2	93600
F	800	0		

迭代一次，80 个集装箱在 C1，40 个集装箱在 E2

共盈利 48000+40×650×4-30×650-15000=117500 元

第五条路径是



路径为 A——B2——C3——D2——E2——F

各个站点	距离	集装箱数	优先满足	运费收入
B2	150	24		7200
C3	500	14	1	14000
D2	650	0		0
E2	650	32	2	416000
F	800	0		0

成本为 15000+(24+14+32)/3×650=30600

盈利为 416000+14000+7200-30600=406600 元

在该题目的假设下，最佳的编组方案是：

1) A——B1——C2——D2——E3——F(共3列,车厢40 40 29)

2) A——B2——C2——D1——E1——F(共1列,车厢40)

3) A——B2——C4——D3——E3——F(共1列,车厢40)

4) A——B1——C1——D1——E2——F(共1列,车厢40)

5) A——B2——C3——D2——E2——F(共1列,车厢24)

总利润为 972000 元。

5.5 问题五

5.5.1 模型的准备

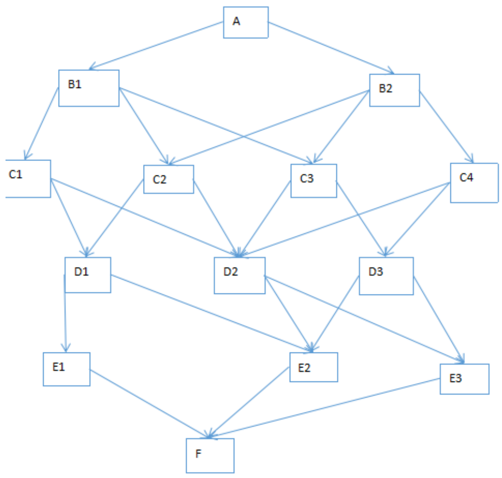
1) 从 A 点到 F 点一共有 24 条路线。
2) 起始点为 A 点，终点为 F 点。
3) 各个站点有相互需求
4) 尽可能满足大多数站对于集装箱的需求，使收益到达最大。

5.5.2 模型的建立

在满足从 A 到各个站点最短路径的前提下，问题五给出了每天各个车站之间潜在的集装箱运输量。对该模型进行改进和优化，综合考虑到 A 点到各个站点的最优路径和各个站点之间的运输量。以站点 A 到站点 B 的距离 d(i, j) 和各个站点的需求 x(i, j) 为变量，改进的线性规划模型如下：

$$Z=\sum_{i=1}^{14}2d(i,j)x(i,j)+Q+\sum_{i=1}^{14}i\times t(i,j)d(i,j)$$

以最短路径为目标函数用 lingo 软件求解：



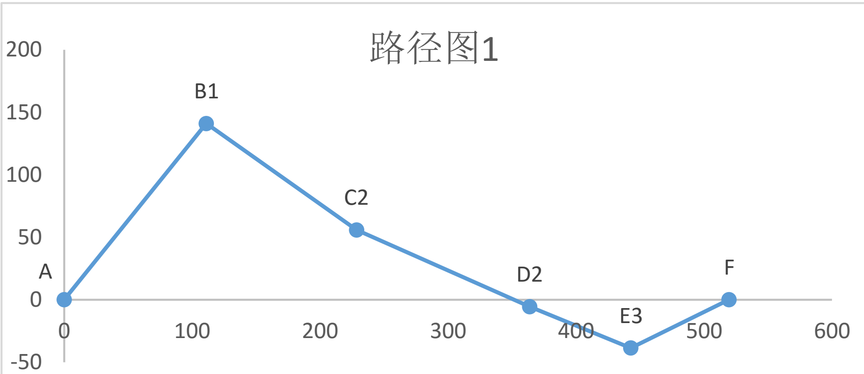
从 A 到 F 点的总路径图

A	B	C	D	E	F	G
路线	T1-T2	T1-T2-T4	T1-T3-T6	T1-T2-T5-T8	T1-T2-T5-T9-T13	T1-T2-T5-T8-T12-T14
距离	250	300	500	550	750	800
路线	T2-T5-T9	T3-T5-T8-T11	T3-T6-T9-T12	T3-T6-T9-T13	T3-T6-T9-T12-T14	T4-T8
距离	400	650	550	600	700	300
路线	T4-T8-T11	T4-T9-T13	T5-T8-T11	T6-T9	T7-T10-T13	
距离	400	500	250	150	350	

各个站点间的最短路径

5.5.3 模型的求解

1.路线一



1-2-5-9-13-14

A-B1-C2-D2—E3-F

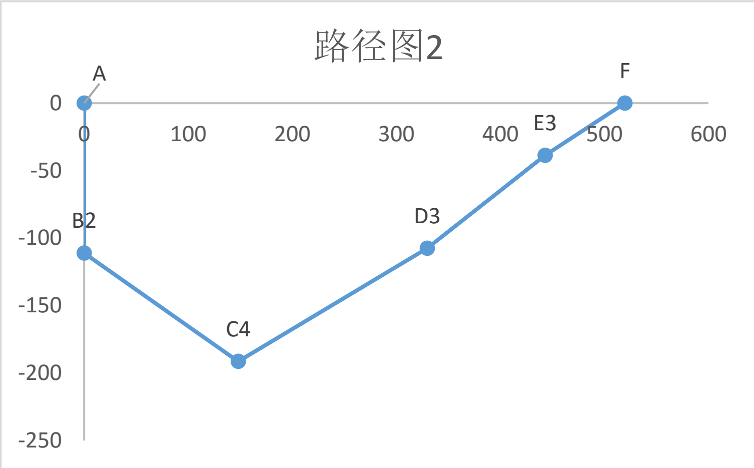
在此路径中,B1 需要 A 提供 57 件集装箱,E3 需要 D2 提供 4 件,F 需要 A 提供 56 件,因此需要一列火车载满 40 个车厢。

P=2×850×56+57×250×2+4×2×100=124500

W=15000+40×850=49000

总盈利为 Z=75500 元

2.路线二



1-3-7-10-13-14

A-B2-C4-D3-E3-F

在此路径中 E3 需要 A、B2、C4 分别提供 44,49,34 件。

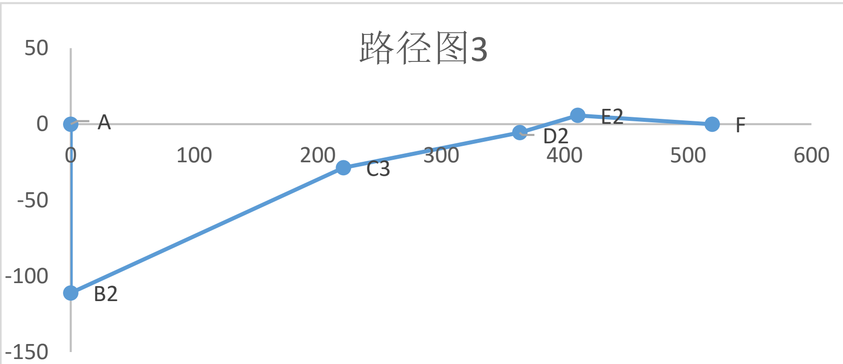
需求量可知超过 120 件,考虑到 2 列火车则不会盈利,则需要 A、B2、C4 分别提供 44,49,27 件

P=44×800×2+650×49×2+34×350×2

W=15000+40×900

Z=P-W=106900 元

3.路线三



1-3-6-9-12-14

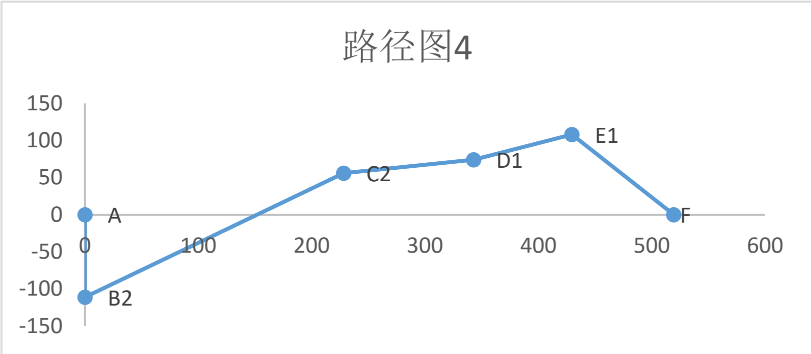
A-B2-C3-D2-E2-F

在此路径中 C3 需要 A 提供 22 件,D2 需要 C3 提供 28 件,E2 需要 B2 提供 57 件,36 节车厢 。P=44×500+114×550+56×150

W=15000+36×650

Z=P-W=54700

4.路线四



1-3-5-8-11-14

A-B2-C2-D1-E1-F

在此路径中需要 2 列火车一列 40 节车厢，一列 37 节车厢。

其中 D1 需要 A 提供 96 件，E1 需要 B2、C2 分别提供 99 件和 19 件，F 需要 B2 提供 15 件。

两列车总 P=700×2×96+650×2×99+250×38+800×30=296600

W=30000+40×950+37×950=103150

Z=P-W=193450 元

路线	A-B1-C2-D2—E3-F	A-B2-C4-D3-E3-F	A-B2-C3-D2-E2-F	A-B2-C2-D1-E1-F
次数(车厢数)	1(40)	1(40)	1(36)	2(40 37)

总结模型

六、模型的评价与改进

6.1 模型的评价

1. 模型优点

1） 要求对多目标进行规划，在充分考虑题目的前提下我们将多规划问题转化为单一目标规划，保证满足各个站点的需求，将目标转化为约束条件大大降低算法的复杂程度。

2） 我们综合考虑到发车的固定成本，和车厢节数的可变成本，从实际问题出发，在第五问，优化线性规划模型，在一定程度上可以降低列车部门的财政花费。

3） 100 天集装箱的变化情况，我们使用了 excel 进行数据处理，能够清晰的得出它为正态分布，方便了后面的概率模型建立。

4） 通过图论的 Dijkstra 算法画出最短路径使复杂的路线清晰可见。

2. 模型缺点

1）本文在考虑各个站点之间的需求量时，并没有考虑到各个站点货物可能有装有卸。

2）在满足各个站点的相互需求时，使用的是实际距离，通过算法算出的最短路径可能不适用。

3）在编组列车时各个路径的选择比较粗糙，并没有深入考虑到多需求，长路径可能带来的经济效益和影响，

参考文献

[1].邰志艳.基于线性规划理论的蔬菜种植技术路线问题建模.[A].吉林：吉林医药学院.2016

[2].严明;刘鸿雁.基于博弈理论的货运列车编组调度多目标优化模型.[J].系统科学学报.2012.1.应用研究

[3].姜启明.数学模型（第四版）[M].北京:高等教育出版社.2011.1

[4].王丽丽.光明市菜篮子工程问题研究[D].山东：山东交通学院.2012.6

附录

附录一：

表 1：货物包装箱相关参数。

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
B	2.22	1.5	1.35	10.5	6
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

表 2：火车车厢相关参数。

车厢类型	长度(米)	宽度	下层高度	上层高度	载重量（吨）
Ⅰ型	12.5	3	2.5	——	55
Ⅱ型	15	3	1.4	1.3	70

表 3：

近 100 天上午集装箱数量：

149	100	106	132	97	102	97	123	124	97
103	130	146	144	108	110	106	133	144	99
128	98	133	101	95	100	144	111	103	106
125	105	112	150	105	144	94	122	148	137
103	140	121	146	148	132	120	115	117	103
93	128	127	137	100	121	149	126	130	144
93	117	95	91	122	125	120	135	98	91
134	107	143	143	146	115	109	139	107	97
111	141	149	112	101	111	131	140	144	130
95	108	139	142	117	115	122	136	129	90

近 100 天下午集装箱数量：

128	137	115	106	133	56	93	95	113	66
155	105	89	108	131	107	98	122	102	102
104	109	106	97	105	87	86	125	124	165
73	82	121	82	119	61	86	113	62	116
73	87	83	136	102	75	106	93	124	97
121	119	103	121	68	84	108	111	92	88
113	85	78	112	90	80	116	75	107	88

92	125	111	91	99	113	98	110	92	80
75	101	85	98	69	61	103	85	112	128
101	102	90	82	111	118	128	88	85	47

表 4：铁路网线说明

铁路网上火车站点名称表：

编号	火车站点名称	X坐标（单位km）	Y坐标（单位km）
1	A	0	0
2	B1	111.1	141.1
3	B2	0.00	-111.1
4	C1	157.9	142.5
5	C2	228.4	55.9
6	C3	220.7	-28.6
7	C4	148.1	-191.4
8	D1	342.9	74.1
9	D2	363.7	-5.6
10	D3	329.7	-107.6
11	E1	429.8	108.1
12	E2	410.7	5.7
13	E3	442.9	-38.6
14	F	519.6	0

站点之间的铁路连结表（直接连接不通过其它站点）：

注意：铁路线路为单向行驶，即火车只能从起点至终点，不能从终点至起点

编号	铁路线起点站点	铁路线终点站点	铁路线长度，即铁路距离(km)
1	A	B1	250
2	A	B2	150
3	B1	C1	50
4	B1	C2	150
5	B1	C3	300
6	B2	C2	400
7	B2	C3	350
8	B2	C4	300
9	C1	D1	300
10	C1	D2	400
11	C2	D1	150

12	C2	D2	250
13	C3	D2	150
14	C3	D3	150
15	C4	D2	400
16	C4	D3	200
17	D1	E1	100
18	D1	E2	100
19	D2	E2	50
20	D2	E3	100
21	D3	E2	150
22	D3	E3	150
23	E1	F	200
24	E2	F	150
25	E3	F	100

表 5：各地集装箱运输需求量（件）

B1	B2	C1	C2	C3	C4	D1	D2	D3	E1	E2	E3	F
58	39	80	54	14	71	82	63	54	23	72	69	72

表 6：各地集装箱运输需求量（件）

起点站	A	A	A	D2	B1	D3	C2	C1	C4	D1	B2	A
终点站	C1	E3	D1	E3	D2	E2	E1	E1	E3	F	F	C3
运输量	10	44	96	4	71	32	19	68	34	22	15	22
起点站	A	C3	C1	D1	A	C1	B2	B2	B2			
终点站	F	D2	D1	E2	B1	E3	E1	E2	E3			
运输量	56	28	30	89	57	93	99	57	49			

附录二：

表 1：

1：Ⅰ型车厢货物可行装载方案 matlab 程序

```
p=[ ];a11=0;
for a11=0:7;
    for a21=0:6;
        for a31=0:5;
            for a41=0:7;
                for a51=0:6;
                    if a21==0;
                        l1=a11*2.81+a31*1.71+a41*2.62+a51*2.53;
```

```
                    end
                    if a21==1,3;5;
 l1=a11*2.81+(a21-1)*1.11+1.5+a31*1.71+a41*2.62+a51*2.53;
                    end
                    if a21==2,4;6;
  l1=a11*2.81+a21*1/2*2.22+a31*1.71+a41*2.62+a51*2.53;
                    end
                    w1=a11*5.5+a21*10.5+a31*9+a41*8+a51*7.5;
                    if l1<=12.5&&w1<=55&&(12.5-l1)<=1.5;
                        p=[p;a11 a21 a31 a41 a51 a11+a21+a31+a41+a51
w1];
                    end
                end
            end
        end
    end
end
p
```

表 2：Ⅰ型车厢货物可行装载方案

A	B	C	D	E	货物总量	货物总质量
0	0	1	0	4	5	39
0	0	1	1	3	5	39.5
0	0	1	2	2	5	40
0	0	1	3	1	5	40.5
0	0	1	4	0	5	41
0	0	2	0	3	5	40.5
0	0	2	1	2	5	41
0	0	2	2	1	5	41.5
0	0	2	3	0	5	42
0	0	4	0	2	6	51
0	0	4	1	1	6	51.5
0	0	4	2	0	6	52
0	0	5	0	1	6	52.5
0	0	5	1	0	6	53
0	1	0	0	4	5	40.5
0	1	0	1	3	5	41
0	1	0	2	2	5	41.5
0	1	0	3	1	5	42
0	1	0	4	0	5	42.5

0	1	1	3	0	5	43.5
0	1	3	0	2	6	52.5
0	1	3	1	1	6	53
0	1	3	2	0	6	53.5
0	2	0	0	4	6	51
0	2	0	1	3	6	51.5
0	2	1	0	3	6	52.5
0	2	1	1	2	6	53
0	2	1	2	1	6	53.5
0	2	1	3	0	6	54
1	0	1	0	3	5	37
1	0	1	1	2	5	37.5
1	0	1	2	1	5	38
1	0	1	3	0	5	38.5
1	0	2	0	2	5	38.5
1	0	2	1	1	5	39
1	0	2	2	0	5	39.5
1	0	4	0	1	6	49
1	0	4	1	0	6	49.5
1	0	5	0	0	6	50.5
1	1	0	0	3	5	38.5
1	1	0	1	2	5	39
1	1	0	2	1	5	39.5
1	1	0	3	0	5	40
1	1	1	0	2	5	40
1	1	1	1	1	5	40.5
1	1	1	2	0	5	41
1	1	3	0	1	6	50.5
1	1	3	1	0	6	51
1	1	4	0	0	6	52
1	2	1	0	2	6	50.5
1	2	1	1	1	6	51
1	2	1	2	0	6	51.5
1	2	2	1	0	6	52.5
2	0	1	0	2	5	35
2	0	1	1	1	5	35.5
2	0	2	0	1	5	36.5
2	0	2	1	0	5	37
2	0	4	0	0	6	47
2	1	0	0	2	5	36.5

2	1	0	1	1	5	37
2	1	0	2	0	5	37.5
2	1	1	0	1	5	38
2	1	1	1	0	5	38.5
2	1	3	0	0	6	48.5
2	2	1	0	1	6	48.5
2	2	1	1	0	6	49
2	2	2	0	0	6	50
3	0	0	1	0	4	24.5
3	0	2	0	0	5	34.5
3	1	0	0	1	5	34.5
3	1	1	0	0	5	36
3	2	1	0	0	6	46.5
4	0	0	0	0	4	22

表 3：Ⅱ型车厢货物可行装载方案 matlab 程序

```
p=[ ];a11=0;
for a11=0:5;
    for a21=0:6;
        for a31=0:7;
            for a41=0:5;
                for a51=0:5;
                    if a21==0;
                        l1=a11*2.81+a31*1.71+a41*2.62+a51*2.53;
                    end
                    if a21==1,3,5;
 l1=a11*2.81+(a21-1)*1.11+1.5+a31*1.71+a41*2.62+a51*2.53;
                    end
                    if a21==2,4,6;
  l1=a11*2.81+a21*1/2*2.22+a31*1.71+a41*2.62+a51*2.53;
                    end
                    if l1<=15&&(15-l1)<=0.2;
                        for a32=0:7;
                            for a42=0:5;
                                for a52=0:5;
                                    l2=1.71*a32+2.62*a42+2.53*a52;
  w2=5.5*a11+10.5*a21+9*(a31+a32)+8*(a41+a42)+7.5*(a51+a52);
                                if l2<=15&&w2<=70;
   p=[p;a11 a21 a31 a41 a51 a32 a42 a52 a11+a21+a31+a41+a51+a32+a42+a52
w2];
                                end
```

```
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
p
```

表 4：Ⅱ型车厢货物可行装载方案

A 下	B 下	C 下	D 下	E 下	C 上	D 上	E 上	货物总量	货物总质量
0	0	1	5	0	0	0	0	6	49
0	0	1	5	0	0	0	1	7	56.5
0	0	1	5	0	0	0	2	8	64
0	0	1	5	0	0	1	0	7	57
0	0	1	5	0	0	1	1	8	64.5
0	0	1	5	0	0	2	0	8	65
0	0	1	5	0	1	0	0	7	58
0	0	1	5	0	1	0	1	8	65.5
0	0	1	5	0	1	1	0	8	66
0	0	1	5	0	2	0	0	8	67
0	2	0	0	5	0	0	0	7	58.5
0	2	0	0	5	0	0	1	8	66
0	2	0	0	5	0	1	0	8	66.5
0	2	0	0	5	1	0	0	8	67.5
0	2	0	1	4	0	0	0	7	59
0	2	0	1	4	0	0	1	8	66.5
0	2	0	1	4	0	1	0	8	67
0	2	0	1	4	1	0	0	8	68
1	0	1	2	2	0	0	0	6	45.5
1	0	1	2	2	0	0	1	7	53
1	0	1	2	2	0	0	2	8	60.5
1	0	1	2	2	1	1	1	9	70
1	0	1	2	2	0	1	0	7	53.5
1	0	1	2	2	0	1	1	8	61
1	0	1	3	1	0	3	0	9	70

1	0	1	2	2	0	2	0	8	61.5
1	0	1	3	1	1	0	2	9	70
1	0	1	4	0	0	2	1	9	70
1	0	1	2	2	1	0	0	7	54.5
1	0	1	2	2	1	0	1	8	62
2	0	4	0	1	0	1	1	9	70
1	0	1	2	2	1	1	0	8	62.5
2	1	0	1	2	2	0	1	9	70
1	0	1	2	2	2	0	0	8	63.5
1	0	1	3	1	0	0	0	6	46
1	0	1	3	1	0	0	1	7	53.5
1	0	1	3	1	0	0	2	8	61
2	1	0	2	1	1	2	0	9	70
1	0	1	3	1	0	1	0	7	54
1	0	1	3	1	0	1	1	8	61.5
2	1	0	3	0	1	1	1	9	70
1	0	1	3	1	0	2	0	8	62
3	2	1	0	1	0	2	0	9	70
3	2	1	1	0	0	1	1	9	70
1	0	1	3	1	1	0	0	7	55
1	0	1	3	1	1	0	1	8	62.5
1	0	1	2	2	0	3	0	9	69.5
1	0	1	3	1	1	1	0	8	63
1	0	1	3	1	2	0	0	8	64
1	0	1	4	0	0	0	0	6	46.5
1	0	1	4	0	0	0	1	7	54
1	0	1	4	0	0	0	2	8	61.5
1	0	1	2	2	1	0	2	9	69.5
1	0	1	4	0	0	1	0	7	54.5
1	0	1	4	0	0	1	1	8	62
1	0	1	3	1	0	2	1	9	69.5
1	0	1	4	0	0	2	0	8	62.5
1	0	1	4	0	0	1	2	9	69.5
1	0	1	4	0	1	0	0	7	55.5
1	0	1	4	0	1	0	1	8	63
1	0	1	4	0	1	1	0	8	63.5
1	0	1	4	0	2	0	0	8	64.5
1	0	4	2	0	0	0	0	7	57.5
1	0	4	2	0	0	0	1	8	65
1	0	4	2	0	0	1	0	8	65.5

1	0	4	2	0	1	0	0	8	66.5
2	0	1	0	3	0	0	0	6	42.5
2	0	1	0	3	0	0	1	7	50
2	0	1	0	3	0	0	2	8	57.5
2	0	1	0	3	3	0	0	9	69.5
2	0	1	0	3	0	1	0	7	50.5
2	0	1	0	3	0	1	1	8	58
2	0	4	0	1	0	0	2	9	69.5
2	0	1	0	3	0	2	0	8	58.5
2	1	0	1	2	1	2	0	9	69.5
2	1	0	2	1	1	1	1	9	69.5
2	0	1	0	3	1	0	0	7	51.5
2	0	1	0	3	1	0	1	8	59
2	1	0	3	0	0	3	0	9	69.5
2	0	1	0	3	1	1	0	8	59.5
2	1	0	3	0	1	0	2	9	69.5
3	2	1	0	1	0	1	1	9	69.5
2	0	1	0	3	2	0	0	8	60.5
3	2	1	1	0	0	0	2	9	69.5
1	0	1	2	2	0	2	1	9	69
1	0	1	3	1	0	1	2	9	69
2	0	4	0	1	0	0	0	7	54.5
2	0	4	0	1	0	0	1	8	62
1	0	1	4	0	0	0	3	9	69
2	0	4	0	1	0	1	0	8	62.5
2	1	0	1	2	1	1	1	9	69
2	0	4	0	1	1	0	0	8	63.5
2	1	0	1	2	0	0	0	6	44.5
2	1	0	1	2	0	0	1	7	52
2	1	0	1	2	0	0	2	8	59.5
2	1	0	2	1	0	3	0	9	69
2	1	0	1	2	0	1	0	7	52.5
2	1	0	1	2	0	1	1	8	60
2	1	0	2	1	1	0	2	9	69
2	1	0	1	2	0	2	0	8	60.5
2	1	0	3	0	0	2	1	9	69
3	1	0	0	2	3	0	0	9	69
2	1	0	1	2	1	0	0	7	53.5
2	1	0	1	2	1	0	1	8	61
3	2	1	0	1	0	0	2	9	69

2	1	0	1	2	1	1	0	8	61.5
1	0	1	2	2	0	1	2	9	68.5
1	0	1	3	1	0	0	3	9	68.5
2	1	0	1	2	2	0	0	8	62.5
2	0	1	0	3	2	1	0	9	68.5
2	1	0	2	1	0	0	0	6	45
2	1	0	2	1	0	0	1	7	52.5
2	1	0	2	1	0	0	2	8	60
2	1	0	1	2	0	3	0	9	68.5
2	1	0	2	1	0	1	0	7	53
2	1	0	2	1	0	1	1	8	60.5
2	1	0	1	2	1	0	2	9	68.5
2	1	0	2	1	0	2	0	8	61
2	1	0	2	1	0	2	1	9	68.5
2	1	0	3	0	0	1	2	9	68.5
2	1	0	2	1	1	0	0	7	54
2	1	0	2	1	1	0	1	8	61.5
1	0	1	2	2	0	0	3	9	68
2	1	0	2	1	1	1	0	8	62
2	0	1	0	3	2	0	1	9	68
2	1	0	1	2	0	2	1	9	68
2	1	0	2	1	2	0	0	8	63
2	1	0	3	0	0	0	0	6	45.5
2	1	0	3	0	0	0	1	7	53
2	1	0	3	0	0	0	2	8	60.5
2	1	0	2	1	0	1	2	9	68
2	1	0	3	0	0	1	0	7	53.5
2	1	0	3	0	0	1	1	8	61
2	1	0	3	0	0	0	3	9	68
2	1	0	3	0	0	2	0	8	61.5
3	1	0	0	2	2	1	0	9	68
2	0	1	0	3	1	2	0	9	67.5
2	1	0	3	0	1	0	0	7	54.5
2	1	0	3	0	1	0	1	8	62
2	1	0	1	2	0	1	2	9	67.5
2	1	0	3	0	1	1	0	8	62.5
2	1	0	2	1	0	0	3	9	67.5
2	1	0	3	0	2	0	0	8	63.5
2	1	3	1	0	0	0	0	7	56.5
2	1	3	1	0	0	0	1	8	64

2	1	3	1	0	0	1	0	8	64.5
2	1	3	1	0	1	0	0	8	65.5
3	1	0	0	2	0	0	0	6	42
3	1	0	0	2	0	0	1	7	49.5
3	1	0	0	2	0	0	2	8	57
3	1	0	0	2	2	0	1	9	67.5
3	1	0	0	2	0	1	0	7	50
3	1	0	0	2	0	1	1	8	57.5
2	0	1	0	3	1	1	1	9	67
3	1	0	0	2	0	2	0	8	58
2	1	0	1	2	0	0	3	9	67
3	1	0	0	2	1	2	0	9	67
3	1	0	0	2	1	0	0	7	51
3	1	0	0	2	1	0	1	8	58.5
2	0	1	0	3	0	3	0	9	66.5
3	1	0	0	2	1	1	0	8	59
2	0	1	0	3	1	0	2	9	66.5
3	1	0	0	2	1	1	1	9	66.5
3	1	0	0	2	2	0	0	8	60
2	0	1	0	3	0	2	1	9	66
3	1	0	0	2	0	3	0	9	66
3	1	0	0	2	1	0	2	9	66
3	2	1	0	1	0	0	0	7	54
3	2	1	0	1	0	0	1	8	61.5
2	0	1	0	3	0	1	2	9	65.5
3	2	1	0	1	0	1	0	8	62
3	1	0	0	2	0	2	1	9	65.5
2	0	1	0	3	0	0	3	9	65
3	2	1	0	1	1	0	0	8	63
3	2	1	1	0	0	0	0	7	54.5
3	2	1	1	0	0	0	1	8	62
3	1	0	0	2	0	1	2	9	65
3	2	1	1	0	0	1	0	8	62.5
3	1	0	0	2	0	0	3	9	64.5
3	2	1	1	0	1	0	0	8	63.5

附录三：

表 1：Ⅰ型车厢货物可行装载方案 matlab 程序

```
p=[];
```

```
for x=0:5;
    for y=0:6;
        for z=0:4;
            if x==3,5;
                s=2.22*[(x-1)/2]+1.5+1.71*y+3*z;
            elseif x==2,4;
                s=2.22*(x/2)+1.71*y+3*z;
            else s=1.5*x+1.71*y+3*z;
            end
            if 10.5*x+9*y+7.5*z<=55&&s<=12.5;
                p=[p;x,y,z];
            end
        end
    end
end
p
```

表 2：Ⅰ型车厢货物可行装载方案（筛选后）

B	C	E
0	0	4
0	1	3
0	2	3
0	3	2
0	4	1
0	5	1
0	6	0
1	0	3
1	1	3
1	2	2
1	3	1
1	4	1
2	0	3
2	1	2
2	2	2
2	3	0
3	0	2
3	1	1
3	2	0
4	0	1
4	1	0
5	0	0

表 3：Ⅱ型车厢货物可行装载方案 matlab 程序

```
p=[];
for x=0:6;
    for y=0:7;
        for z=0:5;
            if x==3,5;
                s=2.22*[(x-1)/2]+1.5+1.71*y+3*z;
            elseif x==2,4;
                s=2.22*(x/2)+1.71*y+3*z;
            else s=1.5*x+1.71*y+3*z;
            end
            if 15-s<=5;
                for y2=0:7;
                    for z2=0:5;
                        s2=1.71*y2+2.53*z2;
                        if s2<=15&&10.5*x+9*(y+y2)+7.5*(z+z2)<=70;
                            p=[p;x y z y2 z2];
                        end
                    end
                end
            end
        end
    end
end
P
```

表 4：Ⅱ型车厢货物可行装载方案（筛选后）

B 下	C 下	D 下	C 上	D 上
0	0	4	0	5
0	0	4	1	4
0	0	4	2	2
0	0	4	3	1
0	0	4	4	0
0	0	5	0	4
0	0	5	1	3
0	0	5	2	1
0	0	5	3	0
0	1	4	0	4
0	1	4	1	2
0	1	4	2	1
0	1	4	3	0
0	1	5	0	3
0	1	5	1	1

0	1	5	2	0
0	2	3	0	3
0	2	3	1	2
0	2	3	2	1
0	2	3	3	0
0	2	4	0	2
0	2	4	1	1
0	2	4	2	0
0	2	5	0	1
0	2	5	1	0
0	3	2	0	3
0	3	2	1	2
0	3	2	2	1
0	3	2	3	0
0	3	3	0	2
0	3	3	1	1
0	3	3	2	0
0	3	4	0	1
0	3	4	1	0
0	3	5	0	0
0	4	2	0	2
0	4	2	1	1
0	4	2	2	0
0	4	3	0	1
0	4	3	1	0
0	4	4	0	0
0	5	1	0	2
0	5	1	1	1
0	5	2	0	1
0	5	2	1	0
0	5	3	0	0
0	6	0	0	2
0	6	0	1	0
0	6	1	0	1
0	6	2	0	0
0	7	0	0	0
1	0	4	0	3
1	0	4	1	2
1	0	4	2	1
1	0	4	3	0

1	0	5	0	2
1	0	5	1	1
1	0	5	2	0
1	1	3	0	3
1	1	3	1	2
1	1	3	2	1
1	1	3	3	0
1	1	4	0	2
1	1	4	1	1
1	1	4	2	0
1	1	5	0	1
1	1	5	1	0
1	2	3	0	2
1	2	3	1	1
1	2	3	2	0
1	2	4	0	1
1	2	4	1	0
1	2	5	0	0
1	3	2	0	2
1	3	2	1	1
1	3	3	0	1
1	3	3	1	0
1	3	4	0	0
1	4	1	0	2
1	4	1	1	0
1	4	2	0	1
1	4	3	0	0
1	5	0	0	1
1	5	0	1	0
1	5	1	0	0
1	6	0	0	0
2	0	4	0	2
2	0	4	1	1
2	0	4	2	0
2	0	5	0	1
2	0	5	1	0
2	1	3	0	2
2	1	3	1	1
2	1	4	0	1
2	1	4	1	0

2	1	5	0	0
2	2	2	0	2
2	2	2	1	0
2	2	3	0	1
2	2	4	0	0
2	3	2	0	0
2	4	1	0	0
2	5	0	0	0
3	0	3	0	2
3	0	3	1	0
3	0	4	0	1
3	0	5	0	0
3	1	2	0	1
3	1	2	1	0
3	1	3	0	0
3	2	2	0	0
3	3	1	0	0
3	4	0	0	0
4	0	2	0	1
4	0	2	1	0
4	0	3	0	0
4	1	1	0	1
4	1	1	1	0
4	1	2	0	0
4	2	1	0	0
4	3	0	0	0
5	0	1	0	1
5	0	1	1	0
5	0	2	0	0
5	1	1	0	0

表 5：求得最优方案 matlab 程序

```
f=ones(1,147);
A=[-1*ones(1,22) ones(1,125)
    -p1(1,:) -p2(1,:)
    -p1(2,:) -p2(2,:) -p2(4,:)
    -p1(3,:) -p2(3,:) -p2(5,:)];
b=[-1-68 -50 -41];
[x,fval]=intlinprog(f,1:147,A,b,[],[],zeros(1,147))
```

改变 b 值
b=[0 -48 -42 -52];

附录四：

表 1：集装箱上午需求量数据拟合 matlab 程序

```
x1=[149 100 106 132 97 102 97 123 124 97];
x2=[103    130 146    144 108 110    106 133 144    99];
x3=[128    98 133 101 95 100 144 111 103 106];
x4=[125 105 112 150 105 144 94 122 148 137];
x5=[103 140 121 146 148 132 120 115 117 103];
x6=[93 128 127 137 100 121 149 126 130 144];
x7=[93 117 95 91 122 125 120 135 98 91];
x8=[134 107 143 143 146 115 109 139 107 97];
x9=[111 141 149 112 101 111 131 140 144 130];
x10=[95 108 139 142 117 115 122 136 129 90];
x=[x1 x2 x3 x4 x5 x6 x7 x8 x9 x10];
normplot(x)
[muhat,sigmahat,muci,sigmaci]=normfit(x)
[h,sig,ci]=ttest(x,120)
```

运行结果：

```
muhat =120.0100
sigmahat =18.1300
muci =116.4126
      123.6074
sigmaci =15.9183
       21.0612
h =0
sig =0.9956
ci =116.4126  123.6074
```

表 2：集装箱下午需求量数据拟合 matlab 程序

```
x1=[128      137  115  106   133   56   93   95    113     66];
x2=[155   105 89   108   131   107   98   122   102   102];
x3=[104      109  106  97     105   87   86   125   124    165];
x4=[73   82   121  82    119   61   86   113   62     116];
x5=[73   87   83   136   102   75   106  93    124    97];
x6=[121      119  103  121   68    84   108  111   92     88];
x7=[113      85   78   112   90    80   116  75    107    88];
x8=[92   125  111  91    99    113  98   110   92     80];
x9=[75   101  85   98    69    61   103  85    112    128];
x10=[101     102  90   82    111   118  128  88    85     47];
x=[x1 x2 x3 x4 x5 x6 x7 x8 x9 x10];
normplot(x)
```

```
[muhat,sigmahat,muci,sigmaci]=normfit(x)
[h,sig,ci]=ttest(x,99.64)
```

运行结果：

```
muhat =99.6400
sigmahat =20.8939
muci =95.4942              103.7858
sigmaci =18.3450             24.2720
```

附录五：

表 1：A 点到各点最短距离及最短路径 matlab 程序

调用了如下 m 文件：

①canshu.m

```
CrossPointNo=14;
for i=1:CrossPointNo
    for j=1:CrossPointNo
        Cost(i,j)=inf;
    end
end

Cost(1,2)=250;Cost(1,3)=150;Cost(2,4)=50;Cost(2,5)=150;Cost(2,6)=300;
Cost(3,5)=400;Cost(3,6)=350;Cost(3,7)=300;
Cost(4,8)=300 ;Cost(4,9)=400;
Cost(5,8)=150 ;Cost( 5,9)=250;Cost(6,9 )=150 ;
Cost(6,10 )=150 ;
Cost( 7,9 )=400 ;
Cost(7 ,10 )=200 ;Cost(8 ,11 )= 100;Cost( 8,12 )=100 ;
Cost( 9,12 )=50 ;Cost( 9,13 )=100 ;
Cost( 10,12 )=150 ;
Cost(10 ,13 )=150 ;Cost( 11,14 )=200;
Cost( 12,14 )= 150;Cost( 13,14 )=100 ;

for i=1:CrossPointNo
    for j=1:CrossPointNo
        if  Cost(i,j) < inf
            Cost(j,i)=Cost(i,j);
        end
    end
end
for i=1:CrossPointNo
    Cost(i,i)=0;
end
```

②printpath.m

```
function y=PrintPath(Min_Distance,Path,StartPointNo,EndPointNo)
              i=EndPointNo;
              if (Path(i)==StartPointNo) & (Min_Distance(i)<Inf)
                  fprintf('起始点(%d)到终止点(%d)的路径为:',StartPointNo,i)
                  fprintf('%d<-',i)
                  fprintf('%d',StartPointNo)
                  fprintf('\n')
              elseif  (Min_Distance(i)==Inf)
                  fprintf('起始点(%d)到终止点(%d)的路径为:空\n',StartPointNo,i)
              else
                   fprintf('起始点(%d)到终止点(%d)点的路径为:',StartPointNo,i)
                   fprintf('%d',i)
                  while(Path(i)~=StartPointNo)
                       fprintf('<-%d', Path(i))
                       i=Path(i);
                   end
                  fprintf('<-%d',StartPointNo)
                   fprintf('\n')
               end
```

③main.m

```
[a b]=ShortestPath_Djk(Cost,CrossPointNo,s);
   PrintPath(a,b,s,e);
  fprintf('路径长度:%f',a(e));
  fprintf('\n');
```

④all.m

```
for i=1:CrossPointNo
    Cost(i,i)=0;
end

Muti_Cost=zeros(CrossPointNo,CrossPointNo);
     for i=1:CrossPointNo
         [a b]=ShortestPath_Djk(Cost,CrossPointNo,i);
         Muti_Cost(i,:)=a;
     End
```

主命令：

```
canshu
s=1,e=14
main
All
```

附录六：

表 1：A 到各点次短路径 matlab 程序

需要调用以下 m 文件：

①canshu.m

②getpath.m

```
function p=getpath(Min_Distance,Path,StartPointNo,EndPointNo)
              i=EndPointNo;np=0;p=[];
              if (Path(i)==StartPointNo) & (Min_Distance(i)<Inf)
                  np=1;p(1)=i;
                  np=2;p(2)=StartPointNo;
                  fprintf('\n')
              elseif  (Min_Distance(i)==Inf)
                  fprintf('起始点(%d)到终止点(%d)的路径为:空
\n',StartPointNo,i)
              else
                  np=1;p(1)=i;
                  while(Path(i)~=StartPointNo)
                       np=np+1;p(np)=Path(i);
                       i=Path(i);
                   end
                  np=np+1;p(np)=StartPointNo;
                   fprintf('\n')
               end



               n=length(p);
               if n~=0
                   q=zeros(1,n);
                   for k=1:n
                        q(k)=p(n+1-k);
                   end
                   p=q;
               End
```

③roadcost.m

```
function y=roadcost(road,c)
y=0;
n=length(road);
```

```
for i=1:(n-1)
    y=y+c(road(i),road(i+1));
end
```

④Shortest_Djk.m

```
function
[Min_Distance,Path]=Shortest_Djk(Cost,CrossPointNo,StartPoint)
for i=1:CrossPointNo
    for j=1:CrossPointNo
        Min_Dist(i,j)=Cost(i,j);
        Muti_Path(i,j)=StartPoint;
        IsFinal(i,j)=0;
    end
end
IsFinal(StartPoint,StartPoint)=1;
   for j=1:(CrossPointNo-1)

       MinPathDist=inf;
        for temp_w=1:CrossPointNo

               if (IsFinal(StartPoint,temp_w)==0) &
(Min_Dist(StartPoint,temp_w)< MinPathDist)
                           temp_v=temp_w;
MinPathDist=Min_Dist(StartPoint,temp_v);
               end


         end
         IsFinal(StartPoint,temp_v)=1;
             for temp_z=1:CrossPointNo
                  if (IsFinal(StartPoint,temp_z)==0)
&( (MinPathDist+Cost(temp_v,temp_z))<(Cost(StartPoint,temp_z)))

Cost(StartPoint,temp_z)=(MinPathDist+Cost(temp_v,temp_z));

Min_Dist(StartPoint,temp_z)=Cost(StartPoint,temp_z);
                           Muti_Path(StartPoint,temp_z)=temp_v;
end
             end
     end
     Min_Distance= Min_Dist(StartPoint,:);
     Path=Muti_Path(StartPoint,:);
```

⑤Shortest.m

```
function p=shortest(startp,endp,Cost)
```

```
CrossPointNo=length(Cost);
[a b]=Shortest_Djk(Cost,CrossPointNo,startp);
  p=getpath(a,b,startp,endp);
  fprintf('路径长度:%f',a(endp));
  fprintf('\n');
```

⑥secshortest.m

```
function sp=secshortest(startp,endp,p,c)
n=length(p);
npp=0;pp=[];
j=-1;
while j~=n-3
    j=j+1;
    c(p(n-j-1),p(n-j))=inf;
    p0=shortest(startp,p(n-j),c);
    n0=length(p0);
    if j==0
        r=[];nr=0;
    else
        nr=nr+1;
        r(nr)=p(n+1-j);
    end
    if r==[]
        npp=1;pp(npp,:)=p0;
    else
        store=p0;
        for i=1:nr
             store(n0+i)=r(nr+1-i);
         end
         npp=npp+1;
         pp(npp,1:length(store))=store;
     end
 end

 np3=0;p3=[];
 for i=1:npp

     if pp(i,length(pp(i,:)))~=0
         l=length(pp(i,:));
     else
         for l=1:length(pp(i,:))
             if pp(i,l)==0
                 break;
             end
         end
```

```
        l=l-1;
    end

    store=pp(i,1:l);
    if roadcost(store,c)>=roadcost(p,c)
        np3=np3+1;
        p3(np3,1:length(store))=store;
    end
end


sp=[];nsp=0;
for i=1:np3
    if p3(i,length(p3(i,:)))~=0
        l=length(p3(i,:));
    else
        for l=1:length(p3(i,:))
            if p3(i,l)==0
                break;
            end
        end
        l=l-1;
    end
    store=p3(i,1:l);

    if i==1
        nsp=1;
        sp(1,1:l)=store;
    else
        if roadcost(store,c)<roadcost(sp(1,:),c)
            sp=[];
            nsp=1;
            sp(1,1:l)=store;
        end
        if roadcost(store,c)==roadcost(sp(1,:),c)
            nsp=nsp+1;
            sp(nsp,1:l)=store;
        end
    end
end
```

主命令：

```
canshu
shortest(1,14,Cost)
```

```
secshortest(1,14,ans,Cost)
```