

1、对象：针对南昌到九江这种类型的开行方案，起始点是南昌，终点是九江的停站方案。中间有四个站点，分别为“永修”、“共青城”、“德安”和“庐山”四个站，总共有 6 个车站（包括起始站点）。

2、路网：采用“A multiobjective model for passenger train services planning: application to Taiwan's high-speed rail line”文献中的数据，7 个站点（1.Taipei 2.Taoyuan 3.Hsinchu 4.Taichung 5.Chiayi 6.Tainan 7.Kaohsiung）。

因此，路网就是一根线，其中，1、4、7 分别为具有接发能力的车站。

3、基础数据：固定的 OD 流量、距离矩阵和时间矩阵。

Table 2
Distance, train running time, and planned hourly passenger volume between stations ($N=7$)

Station i	1. Taipei	2. Taoyuan	3. Hsinchu	4. Taichung	5. Chiayi	6. Tainan	7. Kaohsiung
1. Taipei	(0,0,0)	(35.8,14,683)	(65.6,27,737)	(159.5,40,1407)	(245.1,65,483)	(307.4,83,636)	(338.1,84,2257)
2. Taoyuan	(35.8,14,697)	(0,0,0)	(29.8,11,149)	(123.7,36,748)	(209.3,61,271)	(271.6,79,305)	(302.3,83,861)
3. Hsinchu	(65.6,27,603)	(29.8,11,111)	(0,0,0)	(93.9,24,320)	(179.5,48,53)	(241.8,67,64)	(272.5,71,242)
4. Taichung	(159.5,40,1298)	(123.7,36,731)	(93.9,24,337)	(0,0,0)	(85.6,22,345)	(147.9,40,413)	(178.6,44,900)
5. Chiayi	(245.1,65,340)	(209.3,61,189)	(179.5,48,45)	(85.6,22,246)	(0,0,0)	(62.3,17,187)	(93.0,29,332)
6. Tainan	(307.4,83,513)	(271.6,79,270)	(241.8,67,68)	(147.9,40,295)	(62.3,17,222)	(0,0,0)	(30.7,11,591)
7. Kaohsiung	(338.1,84,2105)	(302.3,83,776)	(272.5,71,241)	(178.6,44,768)	(93.0,29,465)	(30.7,11,817)	(0,0,0)

有疑问：时间矩阵没有看懂，好像有错误。

因此，这里调整了一下，把双向的变成了单向的，而且把时间也修正一下。

distance ($N=7$)

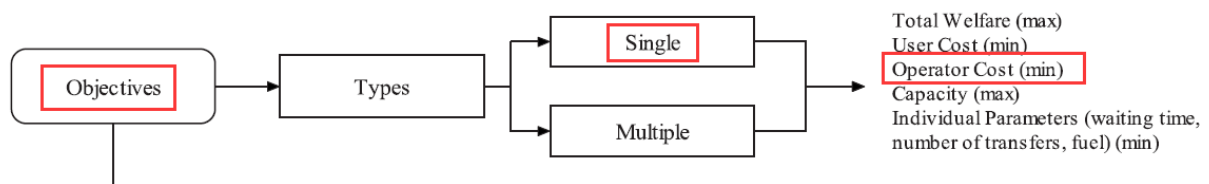
station i	1. Taipei	2. Taoyuan	3. Hsinchu	4. Taichung	5. Chiayi	6. Tainan	7. Kaohsiung
1. Taipei	0	35.8	65.6	159.5	245.1	307.4	338.1
2. Taoyuan	35.8	0	29.8	123.7	209.3	271.6	302.3
3. Hsinchu	65.6	29.8	0	93.9	179.5	241.8	272.5
4. Taichung	159.5	123.7	93.9	0	85.6	147.9	178.6
5. Chiayi	245.1	209.3	179.5	85.6	0	62.3	93
6. Tainan	307.4	271.6	241.8	147.9	62.3	0	30.7
7. Kaohsiung	338.1	302.3	272.5	178.6	93	30.7	0

train running
time ($N=7$)

station i	1. Taipei	2. Taoyuan	3. Hsinchu	4. Taichung	5. Chiayi	6. Tainan	7. Kaohsiung
1. Taipei	0	14	25	49	71	88	99
2. Taoyuan	0	0	11	35	57	74	85
3. Hsinchu	0	0	0	24	46	63	74
4. Taichung	0	0	0	0	22	39	50
5. Chiayi	0	0	0	0	0	17	28
6. Tainan	0	0	0	0	0	0	11

7. Kaohsiung	0	0	0	0	0	0	0
OD (N=7)							
station i	1. Taipei	2. Taoyuan	3. Hsinchu	4. Taichung	5. Chiayi	6. Tainan	7. Kaohsiung
1. Taipei	0	683	737	1407	483	636	2257
2. Taoyuan	697	0	149	748	271	305	861
3. Hsinchu	603	111	0	320	53	64	242
4. Taichung	1298	731	337	0	345	413	900
5. Chiayi	340	189	45	246	0	187	332
6. Tainan	513	270	68	295	222	0	591
7. Kaohsiung	2105	776	241	768	465	817	0

4、目标。目标函数分为两种，单目标和多目标。如下图：



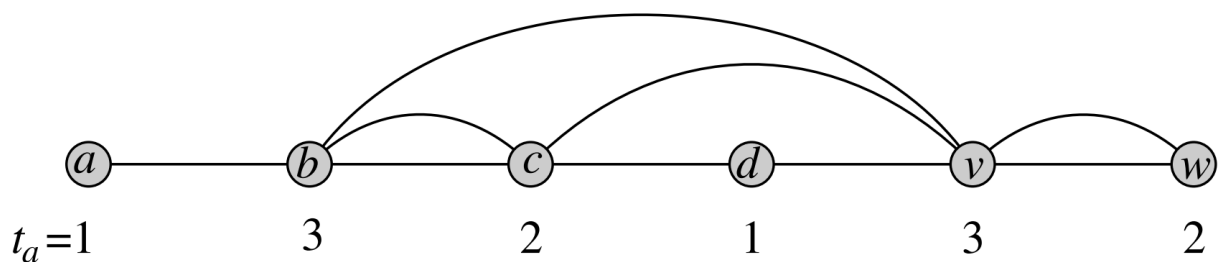
这里选取的单目标是运营成本最小化，运营成本包括固定成本（每开一次的成本 201353）和可变成本（每公里的成本 91459）。

5、决策变量。每个模型的决策变量不同，看下图：



单线选取的决策变量是停站方案（对应网络的是 Routes）和开行频率。

那么，停站方案怎么理解成路径呢？看下图：



如上图所示：方案 1 可以表示成：a-b-v-w；方案 2 可表示成：a-b-c-v-w。

因此，从 a-b 的人有两条路径选择，不同的地方在于时间的不同。

因此，7 个站点，如果用 0 和 1 表示停站/不停站，那么总共有 $2^7=128$

个停站方案，也就是有 128 个开行方案。但是，这些方案当中，比如 0010001、0100100 这些都是可以排除的方案。因此，排除后，总共有 40 个初始方案。这样的话，每一个初始方案都有一个开行频率，那么变量就变成了 $fr(i)$ ， $i=40$ 的数组变量。这样，如果 $fr(5)=0$ ，那么表示第 5 个初始方案排除掉；如果 $fr(25)=3$ ，那么表示优化方案中，第 25 个方案开行 3 趟。

40 个初始方案，用一个 40×7 的矩阵表示，如下：

0	0	0	1	0	0	1
0	0	0	1	0	1	1
0	0	0	1	1	0	1
0	0	0	1	1	1	1
1	0	0	0	0	0	1
1	0	0	0	0	1	1
1	0	0	0	1	0	1
1	0	0	0	1	1	1
1	0	0	1	0	0	0
1	0	0	1	0	0	1
1	0	0	1	0	1	1
1	0	0	1	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	0	1	0	0	1	1
1	0	1	0	1	0	1
1	0	1	0	1	1	1
1	0	1	1	0	0	0
1	0	1	1	0	0	1
1	0	1	1	0	1	1
1	0	1	1	1	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	1
1	1	0	0	0	1	1
1	1	0	0	1	0	1
1	1	0	0	1	1	1
1	1	0	1	0	0	0
1	1	0	1	0	0	1

1	1	0	1	0	1	1
1	1	0	1	1	0	1
1	1	0	1	1	1	1
1	1	1	0	0	0	1
1	1	1	0	0	1	1
1	1	1	0	1	0	1
1	1	1	0	1	1	1
1	1	1	1	0	0	0
1	1	1	1	0	0	1
1	1	1	1	0	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1

6、目标函数的具体表示方式。用以上的变量表示开行方案，那么目标函数可以表示成（lingo）：

```
min = @sum( train(i):201353*Frequency(i) )+
@sum( train(i):91459*Frequency(i)*distance(i) )
```

分析如下：

Varcost=201353：固定成本；

Frequency(i)：每一个 train 的频率；

fixcost=91459：可变成本；

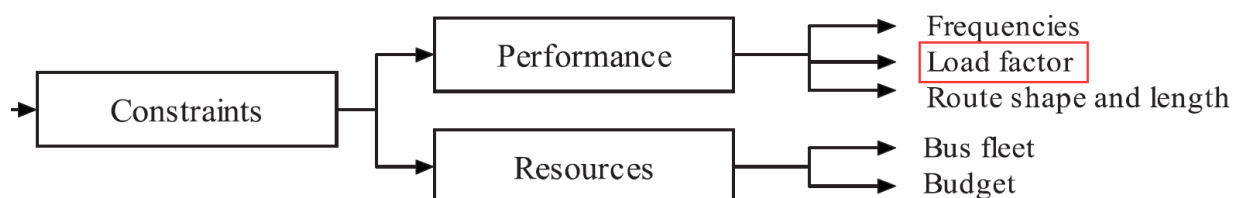
distance(i)：每一个 train 的开行距离；

上面只有 Frequency(i) 是变量数组，distance(i) 是固定的值，是可以求解出来的，根据车站的性质，只有三种方式的距离，分别为：1-4，4-7，1-7 的距离，不管那种开行方案，都是这三种中的一种方式，因此可以用 matlab 将 40 方案的距离求解出来，如下：

0	0	0	1	0	0	1	178.6
0	0	0	1	0	1	1	178.6
0	0	0	1	1	0	1	178.6
0	0	0	1	1	1	1	178.6
1	0	0	0	0	0	1	338.1
1	0	0	0	0	1	1	338.1

1	0	0	0	1	0	1	338.1
1	0	0	0	1	1	1	338.1
1	0	0	1	0	0	0	159.5
1	0	0	1	0	0	1	338.1
1	0	0	1	0	1	1	338.1
1	0	0	1	1	0	1	338.1
1	0	0	1	1	1	1	338.1
1	0	1	0	0	0	1	338.1
1	0	1	0	0	1	1	338.1
1	0	1	0	1	0	1	338.1
1	0	1	0	1	1	1	338.1
1	0	1	1	0	0	0	159.5
1	0	1	1	0	0	1	338.1
1	0	1	1	0	1	1	338.1
1	0	1	1	1	0	1	338.1
1	0	1	1	1	1	1	338.1
1	1	0	0	0	0	1	338.1
1	1	0	0	0	1	1	338.1
1	1	0	0	1	0	1	338.1
1	1	0	0	1	1	1	338.1
1	1	0	1	0	0	0	159.5
1	1	0	1	0	0	1	338.1
1	1	0	1	0	1	1	338.1
1	1	0	1	1	0	1	338.1
1	1	0	1	1	1	1	338.1
1	1	1	0	0	0	1	338.1
1	1	1	0	0	1	1	338.1
1	1	1	0	1	0	1	338.1
1	1	1	0	1	1	1	338.1
1	1	1	1	0	0	0	159.5
1	1	1	1	0	0	1	338.1
1	1	1	1	0	1	1	338.1
1	1	1	1	1	0	1	338.1
1	1	1	1	1	1	1	338.1

7、约束。约束可以有很多种，但是作为单线的开行方案，有些是需要分析的。看下图：

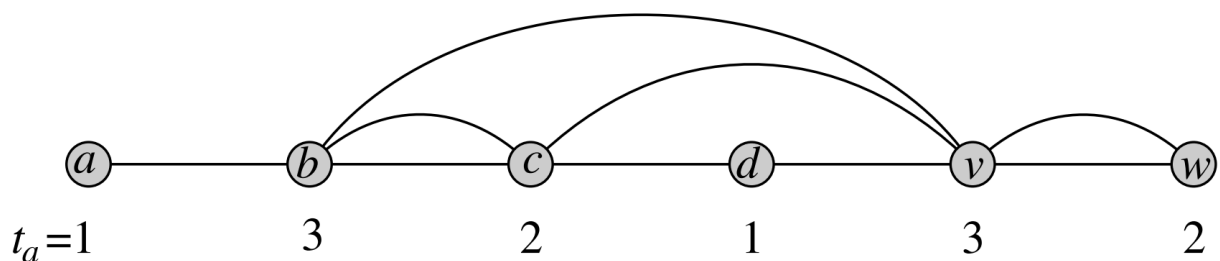


Frequency 代表这整个线路的频率约束，这么说就是所有的频率

数不能大于某个阈值；load factor 代表的路段或者 link 的饱和度和问题，在高铁这块就是某个 link 单位单位时间内可以开行多少车次，也就是频率的问题，这里可以转换成 link_fr 和车站站点 station_fr 问题；Route shape 和 length 可以放在网络中去考虑进去。Fleet 是编组的问题，有具体的数据，也可以考虑一下；budget 是预算的意思，有数据，也可以考虑，但是在我看到的论文里面，好像都没有提到。

8、od 的约束。开行方案的制定要遵循按流开行的原则，因此约束方面，必须要满足带走的 OD 为最大化，也就是必须全部带走 OD。那么这个 OD 怎么表示呢？这里，可以将每一个 od pair 分解到每一个 train 中去，也就是说，7 个站点的 OD 之间的 21 个 od pair 分别分解到 40 个 train 中去，那么设置变量 $\text{Line_passenger}(40 \times 7)$ ，也就是说有 $40 \times 7 = 280$ 个变量，来表示每个 od pair 在每条 train 上的 od 分配。如果 $\text{Line_passenger}(25, 8) = 35$ ，代表编号 25 的 train 带走了编号 8 的 od pair 的人。

9、od 变量的化简。看图：



如上图所示，c-d 或者 a-d 有客流需求，那么按照图上提供的开行方案，是不可能带走的，因为没有有一个 train 在 c 和 d 同时停车，因此 od 变量可以进行化简，就是说，我这边可以设置一个

通过这样的变量设置，就可以将 od 的约束得到。如下：

```
@FOR( train(I):@SUM( od(J): log_line_od( I, J)*line_person( I, J)) <= 800*Frequency(I));  
  
@FOR( od(j):@SUM( train(i): log_line_od( i, j)*line_person( i, j)) = person(j));
```

分析如下：

log_line_od(I, J)：代表的就是0-1变量；

line_person(I, J)：代表的是每条train带走的od；

person(j)：代表的是21个od pair的人数；

因此，从矩阵的每一行看，每次train带走的od加起来不能超过这趟车的定员（定员设置为800）；从矩阵的每一列看，每一个od pair必须被所有的train都带走，因此每个train带走的od pair加起来必须要等于 person（od pair）。

10、总结输入参数：根据以上的两个简单的约束，一个目标函数，其实就可以得到一组优化方案。输入参数总结如下：

一趟车的定员：800（用于约束的判定）；

一趟车的固定成本：201353（用于目标函数的计算）；

一趟车的可变成本：91459（用于目标函数的计算）；

40 趟车的停站形式：40*7 的 0-1 数组（不是输入参数，但是计算其他参数需要）；

40 趟车的行驶距离：40*1 的数组（用于目标函数的计算）；

21 对 od pair：通过 od 矩阵转化过来（就是说二维的化简成一维的）；

Train 和 od pair 的关系矩阵：40*21 的 0-1 数组（表示哪些

train 可以带走哪些 od)；

Train 和 od pair 的 od pair 变量矩阵：40*21 的变量矩阵

（最优化后，可以得到哪些 train 带走了哪些 OD）；

40 趟车的开行频率：40*1 的变量数组；

11、算法。算法主要分成 2 大类：一类是传统的算法

(mathematics programming)，一类是启发式算法

(traditional heuristic or metaheuristic)。对于中

小规模的车行方案，国内的一般应用 metaheuristic 或者

lingo 软件的整数规划（或者混合整数规划）；对于大规模

的车行方案，国外一般用 IBM 的 CPLEX 软件求解，国内用

metaheuristic 求解满意解。

12、结果分析。本例子采用的是 lingo 软件计算，采用的算法是

B-and-B 算法。结果截图如下：

```
Global optimal solution found.
Objective value:           0.3414186E+09
Objective bound:          0.3414186E+09
Infeasibilities:            0.000000
Extended solver steps:        104
Total solver iterations:    12109

Variable      Value      Reduced Cost
FREQUENCY ( TR1)  0.000000    0.1653593E+08
FREQUENCY ( TR2)  0.000000    0.1653593E+08
FREQUENCY ( TR3)  0.000000    0.1653593E+08
FREQUENCY ( TR4)  3.000000    0.1653593E+08
FREQUENCY ( TR5)  0.000000    0.3112364E+08
FREQUENCY ( TR6)  0.000000    0.3112364E+08
FREQUENCY ( TR7)  0.000000    0.3112364E+08
FREQUENCY ( TR8)  1.000000    0.3112364E+08
FREQUENCY ( TR9)  0.000000    0.1478906E+08
FREQUENCY ( TR10) 0.000000    0.3112364E+08
FREQUENCY ( TR11) 0.000000    0.3112364E+08
FREQUENCY ( TR12) 0.000000    0.3112364E+08
FREQUENCY ( TR13) 0.000000    0.3112364E+08
FREQUENCY ( TR14) 1.000000    0.3112364E+08
FREQUENCY ( TR15) 0.000000    0.3112364E+08
FREQUENCY ( TR16) 0.000000    0.3112364E+08
FREQUENCY ( TR17) 0.000000    0.3112364E+08
FREQUENCY ( TR18) 2.000000    0.1478906E+08
FREQUENCY ( TR19) 0.000000    0.3112364E+08
FREQUENCY ( TR20) 0.000000    0.3112364E+08
FREQUENCY ( TR21) 0.000000    0.3112364E+08
FREQUENCY ( TR22) 0.000000    0.3112364E+08
FREQUENCY ( TR23) 0.000000    0.3112364E+08
FREQUENCY ( TR24) 0.000000    0.3112364E+08
FREQUENCY ( TR25) 0.000000    0.3112364E+08
FREQUENCY ( TR26) 1.000000    0.3112364E+08
FREQUENCY ( TR27) 2.000000    0.1478906E+08
FREQUENCY ( TR28) 0.000000    0.3112364E+08
FREQUENCY ( TR29) 0.000000    0.3112364E+08
FREQUENCY ( TR30) 2.000000    0.3112364E+08
FREQUENCY ( TR31) 0.000000    0.3112364E+08
FREQUENCY ( TR32) 0.000000    0.3112364E+08
FREQUENCY ( TR33) 0.000000    0.3112364E+08
FREQUENCY ( TR34) 0.000000    0.3112364E+08
FREQUENCY ( TR35) 0.000000    0.3112364E+08
FREQUENCY ( TR36) 1.000000    0.1478906E+08
FREQUENCY ( TR37) 0.000000    0.3112364E+08
FREQUENCY ( TR38) 1.000000    0.3112364E+08
FREQUENCY ( TR39) 0.000000    0.3112364E+08
FREQUENCY ( TR40) 1.000000    0.3112364E+08
```



整理一下，结果如下：

行车方案编号	行车方案	发车频率
4	4-5-6-7	3
8	1-5-6-7	1
14	1-3-7	1
18	1-3-4	2
26	1-2-5-6-7	1
27	1-2-4	2
30	1-2-4-5-7	2
36	1-2-3-4	1
38	1-2-3-4-6-7	1
40	1-2-3-4-5-6-7	1
	Sum	15

以上就是优化的结果，可以看出，总共开行 10 个开行方案，共 15 次。总的成本是 3.414×10^8 。简单分析，可以看到，居然没有直达车，比如 1-7，1-4 和 4-7，这么说，从成本的角度来说，似乎是尽量少开直达车。调整可变成本和固定成本比例的分析，如果固定成本和可变成本的比值高的话，尽量少开车，调整比例为 2：1，得到如下结果：

Variable	Value	Reduced Cost
FREQUENCY(TR1)	1.000000	180.6000
FREQUENCY(TR2)	0.000000	180.6000
FREQUENCY(TR3)	0.000000	180.6000
FREQUENCY(TR4)	2.000000	180.6000
FREQUENCY(TR5)	0.000000	340.1000
FREQUENCY(TR6)	0.000000	340.1000
FREQUENCY(TR7)	0.000000	340.1000
FREQUENCY(TR8)	1.000000	340.1000
FREQUENCY(TR9)	0.000000	161.5000
FREQUENCY(TR10)	0.000000	340.1000
FREQUENCY(TR11)	0.000000	340.1000
FREQUENCY(TR12)	0.000000	340.1000
FREQUENCY(TR13)	0.000000	340.1000
FREQUENCY(TR14)	1.000000	340.1000
FREQUENCY(TR15)	0.000000	340.1000
FREQUENCY(TR16)	0.000000	340.1000
FREQUENCY(TR17)	0.000000	340.1000
FREQUENCY(TR18)	2.000000	161.5000
FREQUENCY(TR19)	0.000000	340.1000
FREQUENCY(TR20)	0.000000	340.1000
FREQUENCY(TR21)	0.000000	340.1000
FREQUENCY(TR22)	0.000000	340.1000
FREQUENCY(TR23)	0.000000	340.1000
FREQUENCY(TR24)	1.000000	340.1000
FREQUENCY(TR25)	0.000000	340.1000
FREQUENCY(TR26)	0.000000	340.1000
FREQUENCY(TR27)	2.000000	161.5000
FREQUENCY(TR28)	0.000000	340.1000
FREQUENCY(TR29)	0.000000	340.1000
FREQUENCY(TR30)	2.000000	340.1000
FREQUENCY(TR31)	0.000000	340.1000
FREQUENCY(TR32)	0.000000	340.1000
FREQUENCY(TR33)	0.000000	340.1000
FREQUENCY(TR34)	1.000000	340.1000
FREQUENCY(TR35)	0.000000	340.1000
FREQUENCY(TR36)	1.000000	161.5000
FREQUENCY(TR37)	0.000000	340.1000
FREQUENCY(TR38)	0.000000	340.1000
FREQUENCY(TR39)	0.000000	340.1000
FREQUENCY(TR40)	1.000000	340.1000

有了直达车，但是开行的次数为 15，没有变。

据需调整比例为 10：1、1：1 和 1：2、1：10，结果还是一样，开行的次数都是 15。也都没有直达车，还需要继续分析。

13、目标函数的改变。采用旅客时间乘积最小化。广义来说，是乘客成本最小化：包括客票价、等待时间和行程时间（running time and stop time）。由于现在模型是一种类型的车次，因此票价一样，因此不考虑票价，模型考虑的是不转车的乘客数量，因此不需要中转的等待时间，所以就只剩下旅客时间乘积这项指

标，能够直达的（中间不停站的），那是最好。因此需要行程时间的数据。

14、添加输入数据。看时间数据表，如下图：

train running
time (N=7)

station i	1. Taipei	2. Taoyuan	3. Hsinchu	4. Taichung	5. Chiayi	6. Tainan	7. Kaohsiung
1. Taipei	0	14	25	49	71	88	99
2. Taoyuan	0	0	11	35	57	74	85
3. Hsinchu	0	0	0	24	46	63	74
4. Taichung	0	0	0	0	22	39	50
5. Chiayi	0	0	0	0	0	17	28
6. Tainan	0	0	0	0	0	0	11
7. Kaohsiung	0	0	0	0	0	0	0

OD (N=7)

但是，上面显示的是直达（中间不停站）的时间，如果中间停站了，说明需要另外加上停站的时间（这里用了 5 分钟，其实 2-3 分钟就可以了，当然，可以大站 5 分钟，小站 2 分钟）。因此，可以先将 od pair 上了哪趟 train 的时间都算出来，这样每个人的时间都清楚了，得到如下的时间矩阵（40*21），如下表：

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	55	0	0	11
0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	55	0	28	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	44	60	17	33	11	0
0	0	0	0	0	99	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	88	104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11
0	0	0	71	0	104	0	0	0	0	0	0	0	0	0	0	0	0	0	28	0
0	0	0	71	93	109	0	0	0	0	0	0	0	0	0	0	0	0	17	33	11
0	0	49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	49	0	0	104	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0
0	0	49	0	93	109	0	0	0	0	0	0	0	0	0	0	39	55	0	0	11
0	0	49	76	0	109	0	0	0	0	0	0	0	0	22	0	55	0	28	0	0
0	0	49	76	98	114	0	0	0	0	0	0	0	0	22	44	60	17	33	11	0
0	25	0	0	0	104	0	0	0	0	0	0	0	74	0	0	0	0	0	0	0
0	25	0	0	93	109	0	0	0	0	0	0	63	79	0	0	0	0	0	0	11
0	25	0	76	0	109	0	0	0	0	0	46	0	79	0	0	0	0	0	28	0
0	25	0	76	98	114	0	0	0	0	0	46	68	84	0	0	0	17	33	11	0
0	25	54	0	0	0	0	0	0	0	24	0	0	0	0	0	0	0	0	0	0
0	25	54	0	0	109	0	0	0	0	24	0	0	79	0	0	50	0	0	0	0
0	25	54	0	98	114	0	0	0	0	24	0	68	84	0	39	55	0	0	11	0
0	25	54	81	0	114	0	0	0	0	24	51	0	84	22	0	55	0	28	0	0

0	25	54	81	103	119	0	0	0	0	0	24	51	73	89	22	44	60	17	33	11
14	0	0	0	0	104	0	0	0	0	85	0	0	0	0	0	0	0	0	0	0
14	0	0	0	93	109	0	0	0	74	90	0	0	0	0	0	0	0	0	0	11
14	0	0	76	0	109	0	0	57	0	90	0	0	0	0	0	0	0	0	28	0
14	0	0	76	98	114	0	0	57	79	95	0	0	0	0	0	0	0	17	33	11
14	0	54	0	0	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	54	0	0	109	0	35	0	0	90	0	0	0	0	0	0	50	0	0	0
14	0	54	0	98	114	0	35	0	79	95	0	0	0	0	0	39	55	0	0	11
14	0	54	81	0	114	0	35	62	0	95	0	0	0	0	22	0	55	0	28	0
14	0	54	81	103	119	0	35	62	84	100	0	0	0	0	22	44	60	17	33	11
14	30	0	0	0	109	11	0	0	0	90	0	0	0	74	0	0	0	0	0	0
14	30	0	0	98	114	11	0	0	79	95	0	0	63	79	0	0	0	0	0	11
14	30	0	81	0	114	11	0	62	0	95	0	46	0	79	0	0	0	0	28	0
14	30	0	81	103	119	11	0	62	84	100	0	46	68	84	0	0	0	17	33	11
14	30	59	0	0	0	11	40	0	0	0	24	0	0	0	0	0	0	0	0	0
14	30	59	0	0	114	11	40	0	0	95	24	0	0	79	0	0	50	0	0	0
14	30	59	0	103	119	11	40	0	84	100	24	0	68	84	0	39	55	0	0	11
14	30	59	86	0	119	11	40	67	0	100	24	51	0	84	22	0	55	0	28	0
14	30	59	86	108	124	11	40	67	89	105	24	51	73	89	22	44	60	17	33	0;

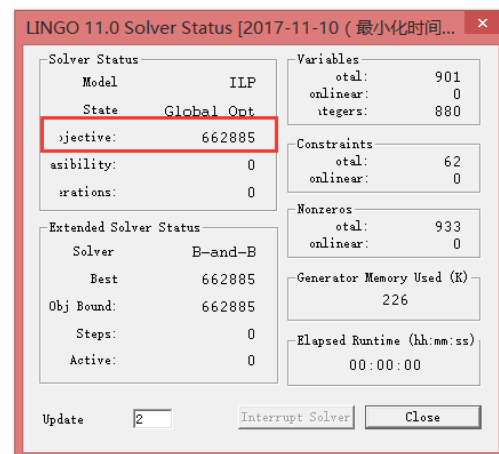
得到了时间矩阵后，就可以得到目标函数了，看 lingo 代码：

```
min = @sum( rel: line_od_time*line_person);
```

最后的结果是：

```
Global optimal solution found.
Objective value:                662885.0
Objective bound:                662885.0
Infeasibilities:                0.000000
Extended solver steps:          0
Total solver iterations:        0
```

Variable	Value	Reduced Cost
FREQUENCY (TR1)	2.000000	0.000000
FREQUENCY (TR2)	3.000000	0.000000
FREQUENCY (TR3)	2.000000	0.000000
FREQUENCY (TR4)	4.000000	0.000000
FREQUENCY (TR5)	3.000000	0.000000
FREQUENCY (TR6)	5.000000	0.000000
FREQUENCY (TR7)	4.000000	0.000000
FREQUENCY (TR8)	6.000000	0.000000
FREQUENCY (TR9)	2.000000	0.000000
FREQUENCY (TR10)	6.000000	0.000000
FREQUENCY (TR11)	8.000000	0.000000
FREQUENCY (TR12)	8.000000	0.000000
FREQUENCY (TR13)	10.00000	0.000000
FREQUENCY (TR14)	5.000000	0.000000
FREQUENCY (TR15)	6.000000	0.000000
FREQUENCY (TR16)	6.000000	0.000000
FREQUENCY (TR17)	7.000000	0.000000
FREQUENCY (TR18)	4.000000	0.000000
FREQUENCY (TR19)	8.000000	0.000000
FREQUENCY (TR20)	10.00000	0.000000
FREQUENCY (TR21)	9.000000	0.000000
FREQUENCY (TR22)	12.00000	0.000000
FREQUENCY (TR23)	5.000000	0.000000
FREQUENCY (TR24)	7.000000	0.000000
FREQUENCY (TR25)	7.000000	0.000000
FREQUENCY (TR26)	9.000000	0.000000
FREQUENCY (TR27)	4.000000	0.000000
FREQUENCY (TR28)	9.000000	0.000000
FREQUENCY (TR29)	12.00000	0.000000
FREQUENCY (TR30)	11.00000	0.000000
FREQUENCY (TR31)	14.00000	0.000000
FREQUENCY (TR32)	7.000000	0.000000
FREQUENCY (TR33)	9.000000	0.000000
FREQUENCY (TR34)	8.000000	0.000000
FREQUENCY (TR35)	10.00000	0.000000
FREQUENCY (TR36)	6.000000	0.000000
FREQUENCY (TR37)	11.00000	0.000000
FREQUENCY (TR38)	13.00000	0.000000
FREQUENCY (TR39)	13.00000	0.000000
FREQUENCY (TR40)	15.00000	0.000000



可以看到，目标值为662885，但是解很离谱，每个方案都开，说明是

不计成本。分析了train和od的数据，发现有的车跑空车，说明lingo有多重解的时候，只显示一种解。因此，添加约束，给开行的train添加一个上座率约束，必选大于某个阈值。通过不同的阈值选取，可以得到不同的最优解。

当每个train的人数要求 ≥ 500 ，目标值不变662885

```
Global optimal solution found.
Objective value:           662885.0
Objective bound:          662885.0
Infeasibilities:          0.000000
Extended solver steps:    0
Total solver iterations:  24808
```

Variable	Value	Reduced Cost
FREQUENCY (TR1)	1.000000	0.000000
FREQUENCY (TR2)	0.000000	0.000000
FREQUENCY (TR3)	1.000000	0.000000
FREQUENCY (TR4)	0.000000	0.000000
FREQUENCY (TR5)	3.000000	0.000000
FREQUENCY (TR6)	1.000000	0.000000
FREQUENCY (TR7)	1.000000	0.000000
FREQUENCY (TR8)	0.000000	0.000000
FREQUENCY (TR9)	2.000000	0.000000
FREQUENCY (TR10)	0.000000	0.000000
FREQUENCY (TR11)	1.000000	0.000000
FREQUENCY (TR12)	0.000000	0.000000
FREQUENCY (TR13)	0.000000	0.000000
FREQUENCY (TR14)	0.000000	0.000000
FREQUENCY (TR15)	1.000000	0.000000
FREQUENCY (TR16)	1.000000	0.000000
FREQUENCY (TR17)	0.000000	0.000000
FREQUENCY (TR18)	0.000000	0.000000
FREQUENCY (TR19)	1.000000	0.000000
FREQUENCY (TR20)	0.000000	0.000000
FREQUENCY (TR21)	0.000000	0.000000
FREQUENCY (TR22)	0.000000	0.000000
FREQUENCY (TR23)	2.000000	0.000000
FREQUENCY (TR24)	1.000000	0.000000
FREQUENCY (TR25)	1.000000	0.000000
FREQUENCY (TR26)	0.000000	0.000000
FREQUENCY (TR27)	1.000000	0.000000
FREQUENCY (TR28)	0.000000	0.000000
FREQUENCY (TR29)	0.000000	0.000000
FREQUENCY (TR30)	0.000000	0.000000
FREQUENCY (TR31)	0.000000	0.000000
FREQUENCY (TR32)	1.000000	0.000000
FREQUENCY (TR33)	0.000000	0.000000
FREQUENCY (TR34)	0.000000	0.000000
FREQUENCY (TR35)	0.000000	0.000000
FREQUENCY (TR36)	0.000000	0.000000
FREQUENCY (TR37)	0.000000	0.000000
FREQUENCY (TR38)	0.000000	0.000000
FREQUENCY (TR39)	0.000000	0.000000
FREQUENCY (TR40)	1.000000	0.000000

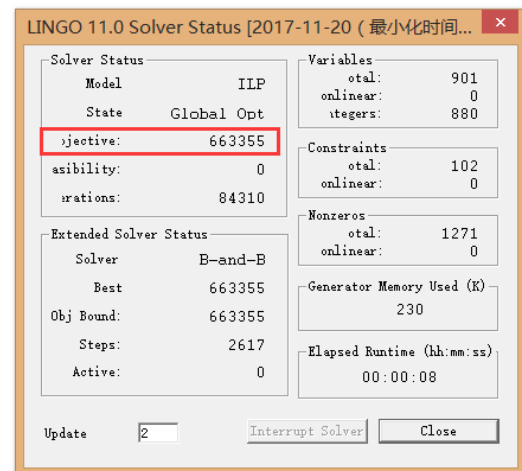


行车方案编号	行车方案	发车频率
1	4-7	1
3	4-5-7	1
5	1-7	3
6	1-6-7	1
7	1-5-7	1
9	1-4	2
11	1-4-6-7	1
15	1-3-6-7	1
16	1-3-5-7	1
19	1-3-4-7	1
23	1-2-7	2
24	1-2-6-7	1
25	1-2-5-7	1
27	1-2-4	1
32	1-2-3-7	1
40	1-2-3-4-5-6-7	1
	sum	20

当每个train的人数要求 ≥ 600 ，目标值663355

```
Global optimal solution found.
Objective value:           663355.0
Objective bound:           663355.0
Infeasibilities:            0.000000
Extended solver steps:      2617
Total solver iterations:    84310
```

Variable	Value	Reduced Cost
FREQUENCY (TR1)	1.000000	0.000000
FREQUENCY (TR2)	0.000000	0.000000
FREQUENCY (TR3)	0.000000	0.000000
FREQUENCY (TR4)	0.000000	0.000000
FREQUENCY (TR5)	3.000000	0.000000
FREQUENCY (TR6)	1.000000	0.000000
FREQUENCY (TR7)	0.000000	0.000000
FREQUENCY (TR8)	1.000000	0.000000
FREQUENCY (TR9)	2.000000	0.000000
FREQUENCY (TR10)	0.000000	0.000000
FREQUENCY (TR11)	0.000000	0.000000
FREQUENCY (TR12)	0.000000	0.000000
FREQUENCY (TR13)	0.000000	0.000000
FREQUENCY (TR14)	0.000000	0.000000
FREQUENCY (TR15)	1.000000	0.000000
FREQUENCY (TR16)	1.000000	0.000000
FREQUENCY (TR17)	0.000000	0.000000
FREQUENCY (TR18)	0.000000	0.000000
FREQUENCY (TR19)	0.000000	0.000000
FREQUENCY (TR20)	0.000000	0.000000
FREQUENCY (TR21)	1.000000	0.000000
FREQUENCY (TR22)	0.000000	0.000000
FREQUENCY (TR23)	1.000000	0.000000
FREQUENCY (TR24)	1.000000	0.000000
FREQUENCY (TR25)	0.000000	0.000000
FREQUENCY (TR26)	1.000000	0.000000
FREQUENCY (TR27)	0.000000	0.000000
FREQUENCY (TR28)	1.000000	0.000000
FREQUENCY (TR29)	1.000000	0.000000
FREQUENCY (TR30)	0.000000	0.000000
FREQUENCY (TR31)	0.000000	0.000000
FREQUENCY (TR32)	1.000000	0.000000
FREQUENCY (TR33)	0.000000	0.000000
FREQUENCY (TR34)	0.000000	0.000000
FREQUENCY (TR35)	0.000000	0.000000
FREQUENCY (TR36)	0.000000	0.000000
FREQUENCY (TR37)	0.000000	0.000000
FREQUENCY (TR38)	0.000000	0.000000
FREQUENCY (TR39)	0.000000	0.000000
FREQUENCY (TR40)	1.000000	0.000000



统计分析一下：编号 1，5，9 的直达 train 存在，符合预期。总共发车 18 次。

当每个 train 的人数要求 ≥ 400 ，目标值 662885，说明上座率小于某个数的时候，目标值是不变的，但是解是会变的。

因此，可以考虑一个约束，上座率的约束，就是为了保证左右的乘客不转车，但是又最大化 train 的上座率。依此引申到 train 的空座时间积等概念，就是尽量保证没有空位等目标。

15、多目标规划。2 种方法进行求解，一种是转变成目标规划的方法，设立级别优先，最小化偏差+和偏差-进行求解；一种是直接给两个目标添加权重系数，根据目标的偏重洗好进行求解。

16、采用目标规划方法求解。用类似以下公式求解：

$$\min Z = P_1 d_1^- + P_2 (d_2^- + d_3^-) + P_3 d_4^+ + P_4 d_5^-, \text{ 其中 } P_1, P_2, P_3, P_4 \text{ 均为优先级.}$$

由于目标为 2 个目标，都是最小化求解。因此采用如下所示：

```
min = d1plus+100*d2plus;
```

柔性约束为：

```
@sum( train(i):201353*Frequency(i) )+@sum(
train(i):91459*Frequency(i)*distance(i) )+d
lminus-d1plus = 3.414*10^8 ;
```

```
@sum( rel:
line_od_time*line_person)+d2minus-d2plus =
662885;
```

得到如下结果：

D1PLUS	18595.00
D2PLUS	19036.00

那么目标 1 的最优值为 $3.414 \times 10^8 + 18595$ ；

那么目标 2 的最优值为 $662885 + 19036$ ；

但是解的分析可以看出：三种直达车都有，说明更符合预判，但是没有站站停的车，截图如下：

Variable	Value	Reduced Cost
D1PLUS	18595.00	0.000000
D2PLUS	19036.00	0.000000
D1MINUS	0.000000	1.000000
D2MINUS	0.000000	100.0000
FREQUENCY (TR1)	1.000000	0.1653593E+08
FREQUENCY (TR2)	1.000000	0.1653593E+08
FREQUENCY (TR3)	0.000000	0.1653593E+08
FREQUENCY (TR4)	1.000000	0.1653593E+08
FREQUENCY (TR5)	2.000000	0.3112364E+08
FREQUENCY (TR6)	1.000000	0.3112364E+08
FREQUENCY (TR7)	1.000000	0.3112364E+08
FREQUENCY (TR8)	0.000000	0.3112364E+08
FREQUENCY (TR9)	1.000000	0.1478906E+08
FREQUENCY (TR10)	0.000000	0.3112364E+08
FREQUENCY (TR11)	0.000000	0.3112364E+08
FREQUENCY (TR12)	0.000000	0.3112364E+08
FREQUENCY (TR13)	0.000000	0.3112364E+08
FREQUENCY (TR14)	1.000000	0.3112364E+08
FREQUENCY (TR15)	0.000000	0.3112364E+08
FREQUENCY (TR16)	0.000000	0.3112364E+08
FREQUENCY (TR17)	0.000000	0.3112364E+08
FREQUENCY (TR18)	1.000000	0.1478906E+08
FREQUENCY (TR19)	0.000000	0.3112364E+08
FREQUENCY (TR20)	0.000000	0.3112364E+08
FREQUENCY (TR21)	0.000000	0.3112364E+08
FREQUENCY (TR22)	0.000000	0.3112364E+08
FREQUENCY (TR23)	1.000000	0.3112364E+08
FREQUENCY (TR24)	0.000000	0.3112364E+08
FREQUENCY (TR25)	0.000000	0.3112364E+08
FREQUENCY (TR26)	0.000000	0.3112364E+08
FREQUENCY (TR27)	2.000000	0.1478906E+08
FREQUENCY (TR28)	0.000000	0.3112364E+08
FREQUENCY (TR29)	0.000000	0.3112364E+08
FREQUENCY (TR30)	0.000000	0.3112364E+08
FREQUENCY (TR31)	0.000000	0.3112364E+08
FREQUENCY (TR32)	0.000000	0.3112364E+08
FREQUENCY (TR33)	0.000000	0.3112364E+08
FREQUENCY (TR34)	0.000000	0.3112364E+08
FREQUENCY (TR35)	1.000000	0.3112364E+08
FREQUENCY (TR36)	1.000000	0.1478906E+08
FREQUENCY (TR37)	0.000000	0.3112364E+08
FREQUENCY (TR38)	0.000000	0.3112364E+08
FREQUENCY (TR39)	0.000000	0.3112364E+08
FREQUENCY (TR40)	0.000000	0.3112364E+08
DISTANCE (TR1)	178.6000	0.000000

17、更换参数。将运营费用的系数调整，得到最优解的量纲级别变成和“662885”同样的级别。调整系数为 400 和 180，得到最小值“672000”。然后再重新目标规划一次，只是目标约束（柔性约束）变成以下的：

! 柔性约束；

```
@sum( train(i):400*Frequency(i) )+@sum( train(i):1  
80*Frequency(i)*distance(i) )+d1minus-d1plus =  
672000;  
  
@sum( rel: line_od_time*line_person)+d2minus-d2plus  
= 662885;
```

目标函数为：

```
min = 100*d1plus+d2plus;
```

运营目标等级比旅客时间等级要高

结果如下：

D1PLUS	0.000000
D2PLUS	19036.00

说明第一目标还是最优值，但是解变了，也就是说第一目标是有多个解的；（但目标有站站停方案，有一个 4-7 的直达方案）

第二目标比最优值大了 19036；

从解的分析可以看出：三种直达车都有，符合预期。但是也还是没有站站停的 train。

Variable	Value	Reduced Cost
D1PLUS	0.000000	0.000000
D2PLUS	19036.00	0.000000
D1MINUS	0.000000	100.0000
D2MINUS	0.000000	1.000000
FREQUENCY (TR1)	1.000000	3254800.
FREQUENCY (TR2)	1.000000	3254800.
FREQUENCY (TR3)	0.000000	3254800.
FREQUENCY (TR4)	1.000000	3254800.
FREQUENCY (TR5)	2.000000	6125800.
FREQUENCY (TR6)	1.000000	6125800.
FREQUENCY (TR7)	1.000000	6125800.
FREQUENCY (TR8)	0.000000	6125800.
FREQUENCY (TR9)	1.000000	2911000.
FREQUENCY (TR10)	0.000000	6125800.
FREQUENCY (TR11)	0.000000	6125800.
FREQUENCY (TR12)	0.000000	6125800.
FREQUENCY (TR13)	0.000000	6125800.
FREQUENCY (TR14)	1.000000	6125800.
FREQUENCY (TR15)	0.000000	6125800.
FREQUENCY (TR16)	0.000000	6125800.
FREQUENCY (TR17)	0.000000	6125800.
FREQUENCY (TR18)	2.000000	2911000.
FREQUENCY (TR19)	0.000000	6125800.
FREQUENCY (TR20)	0.000000	6125800.
FREQUENCY (TR21)	0.000000	6125800.
FREQUENCY (TR22)	0.000000	6125800.
FREQUENCY (TR23)	1.000000	6125800.
FREQUENCY (TR24)	0.000000	6125800.
FREQUENCY (TR25)	0.000000	6125800.
FREQUENCY (TR26)	0.000000	6125800.
FREQUENCY (TR27)	1.000000	2911000.
FREQUENCY (TR28)	0.000000	6125800.
FREQUENCY (TR29)	0.000000	6125800.
FREQUENCY (TR30)	0.000000	6125800.
FREQUENCY (TR31)	0.000000	6125800.
FREQUENCY (TR32)	0.000000	6125800.
FREQUENCY (TR33)	0.000000	6125800.
FREQUENCY (TR34)	0.000000	6125800.
FREQUENCY (TR35)	1.000000	6125800.
FREQUENCY (TR36)	1.000000	2911000.
FREQUENCY (TR37)	0.000000	6125800.
FREQUENCY (TR38)	0.000000	6125800.
FREQUENCY (TR39)	0.000000	6125800.
FREQUENCY (TR40)	0.000000	6125800.

调整目标等级，将旅客时间放在第一等级：

$\min = 100 * d2plus + d1plus;$

结果如下：

Variable	Value	Reduced Cost
D2PLUS	890.0000	0.000000
D1PLUS	215122.0	0.000000
D1MINUS	0.000000	1.000000
D2MINUS	0.000000	100.0000
FREQUENCY (TR1)	1.000000	32548.00
FREQUENCY (TR2)	0.000000	32548.00
FREQUENCY (TR3)	0.000000	32548.00
FREQUENCY (TR4)	0.000000	32548.00
FREQUENCY (TR5)	3.000000	61258.00
FREQUENCY (TR6)	1.000000	61258.00
FREQUENCY (TR7)	0.000000	61258.00
FREQUENCY (TR8)	1.000000	61258.00
FREQUENCY (TR9)	1.000000	29110.00
FREQUENCY (TR10)	1.000000	61258.00
FREQUENCY (TR11)	0.000000	61258.00
FREQUENCY (TR12)	0.000000	61258.00
FREQUENCY (TR13)	0.000000	61258.00
FREQUENCY (TR14)	1.000000	61258.00
FREQUENCY (TR15)	0.000000	61258.00
FREQUENCY (TR16)	0.000000	61258.00
FREQUENCY (TR17)	0.000000	61258.00
FREQUENCY (TR18)	0.000000	29110.00
FREQUENCY (TR19)	0.000000	61258.00
FREQUENCY (TR20)	1.000000	61258.00
FREQUENCY (TR21)	0.000000	61258.00
FREQUENCY (TR22)	0.000000	61258.00
FREQUENCY (TR23)	1.000000	61258.00
FREQUENCY (TR24)	1.000000	61258.00
FREQUENCY (TR25)	1.000000	61258.00
FREQUENCY (TR26)	0.000000	61258.00
FREQUENCY (TR27)	1.000000	29110.00
FREQUENCY (TR28)	0.000000	61258.00
FREQUENCY (TR29)	0.000000	61258.00
FREQUENCY (TR30)	0.000000	61258.00
FREQUENCY (TR31)	0.000000	61258.00
FREQUENCY (TR32)	0.000000	61258.00
FREQUENCY (TR33)	0.000000	61258.00
FREQUENCY (TR34)	0.000000	61258.00
FREQUENCY (TR35)	0.000000	61258.00
FREQUENCY (TR36)	0.000000	29110.00
FREQUENCY (TR37)	0.000000	61258.00
FREQUENCY (TR38)	0.000000	61258.00
FREQUENCY (TR39)	1.000000	61258.00
FREQUENCY (TR40)	1.000000	61258.00

18、总结：目标和约束都是最简单的形式，变量的设置尤为关键，采用何种数据结构比较重要。这里都是先用 matlab 做数据的预处理，得到输入参数，然后用 lingo 直接求解线性规划模型，从软件的运行可以看出，用的是 B-and-B 的方法，针对 1000 多个变量，也是在 1 分钟之内，但是如果问题再大的话，就要 20 分钟才能算出来。

19、后期继续。

增加约束：将边和点的约束点考虑进去，由于是直线的开行方案，并没

有路径的选择，因此边和点的约束并没有关系，但是路网就有这个约束；将方案总的开行趟数加进去；将预算加进去；将 train 的上座率加进去。

增加目标：整个路网的平衡（考虑自己定义一个平衡），道路网考虑的小汽车的路网平衡，train 的 od 是人，道路网的 od 就是小车，因此是不一样的，因此直接考虑 train 的路网平衡性能；train 的空载运行时间（这个还没有仔细考虑）等。当然，约束和目标可以互换，考虑模型的可变性。

算法：现在用的是 matlab+lingo 软件求解，针对网络开行方案想看看用启发式的算法得到满意解，但是有两个方面困扰，一个是变量的表示，一个是 od 的配流（应该是配到了 train 上面，还是配到了网络图上），怎么评价配流的效果。如果用一种算法得到后，其他的问题就比较好办了。

软件的应用：现在用 lingo 求解的时候，针对大规模的问题，还是有点困难（其实也不会太困难，毕竟是规划，要计算的时候，一个晚上运算我觉的可以了），想用 ibm 的 Cplex 软件试试求解（已经下载了破解版本），这个软件很牛，只能线性，但是速度快（lingo10 分的话，他有 50 分）。

小论文：小论文希望写的时候模型能够复杂化（是否再添加其他因素进去，比如 train 的类型 D 和 G；模型采用多目标，将目标再改进一些）；算例的数据不知道用新的数据，还是其他论文有的数据；结果分析的时候希望能够多分析一下（有人协助）。

19、问题总结。

可变成本和固定成本：

表 2 各类列车参数表
Table 2 Table about all kinds of trains

种型	A	B	C
动车组(辆)	3	2	1
列车长度(辆)	6	8	16
定员(人/辆)	80	80	128
C_1 (元/辆)	4 000	4 000	3 000
C_2 (元/辆)	10	10	8
U_1 (元/辆km)	10 000	10 000	8 000
U_2 (元/辆km)	20	20	15
速度(km/h)	350/320/300	250/220/200	200/180/160
运价率(元/人km)	1/0.9/0.8	0.9/0.8/0.7	0.8/0.7/0.6
停站时间(min)	4/4/0	8/6/0	15/12/10
换乘时间(min)	4/6/8	6/8/10	8/10/12
整备时间(min)	15	20	30

表 2 2种高速列车的广义费用函数参数

列车 种类	单位票价 /元	速度 /(km/h)	容量 /人	固定成本 /(元/车次)	可变成本 /(元/车次公里)
A	0.5	300	600	1 000	30
B	0.4	250	600	800	25

Table 1
Input parameters of the model

Parameter	Value (NT\$: New Taiwan Dollar) (US\$1 \approx NT\$32)
Set of stations (Ω)	{1,2,3,4,5,6,7}
Set of shunting or terminal stations (Φ)	{1,4,7}
Operating hours (H_i)	1 h
Terminal time required (G_r)	45 min/train (round trip)
Train seating capacity (Q_{tr})	800 seats/train
Service line and station capacity (E)	15 trains/h
Extra time required for stops at stations (W_i)	3 min/station
Fixed overhead cost (C_1)	NT\$201353/train-day
Variable operating cost (C_2)	NT\$91459/train-km

表 1 各类列车参数

	G 列车	D 列车
定员(人)	500	550
固定成本(元)	10000	6000
可变成本(元/km)	20	15
最大速度(km/h)	350	300

中转客流：很多文本提到了中转客流，但是我看了半天，没有看懂，怎么识别是否中转。

配流：如果采用乘客 od 直接配流的方式，和开行方案的关系还是理不清楚。