图论中 TSP 问题及 LINGO 求解技巧

巡回旅行商问题(Traveling Salesman Problem，TSP)，也称为货郎担问题。最早可以追溯到1759年Euler提出的骑士旅行问题。1948年，由美国兰德公司推动，TSP成为近代组合优化领域的一个典型难题。它已经被证明属于NP难题。

用图论描述TSP，给出一个图 $G=(V,E)$，每边 $e\in E$ 上有非负权值 $w(e)$，寻找G的Hamilton圈C，使得C的总权 $W(C)=\sum\limits_{e\in E(C)} w(e)$ 最小.

几十年来，出现了很多近似优化算法。如近邻法、贪心算法、最近插入法、最远插入法、模拟退火算法以及遗传算法。这里我们介绍利用LINGO软件进行求解的方法。

问题1　设有一个售货员从10个城市中的某一个城市出发，去其它9个城市推销产品。10个城市相互距离如下表。要求每个城市到达一次仅一次后，回到原出发城市。问他应如何选择旅行路线，使总路程最短。

表1　10个城市距离表

城市	1	2	3	4	5	6	7	8	9	10
1	0	7	4	5	8	6	12	13	11	18
2	7	0	3	10	9	14	5	14	17	17
3	4	3	0	5	9	10	21	8	27	12
4	5	10	5	0	14	9	10	9	23	16
5	8	9	9	14	0	7	8	7	20	19
6	6	14	10	9	7	0	13	5	25	13
7	12	5	21	10	8	13	0	23	21	18
8	13	14	8	9	7	5	23	0	18	12
9	11	17	27	23	20	25	21	18	0	16
10	18	17	12	16	19	13	18	12	16	0

我们采用线性规划的方法求解

设城市之间距离用矩阵 d 来表示，d_{ij} 表示城市 i 与城市 j 之间的距离。设0--1矩阵 X 用来表示经过的各城市之间的路线。设

$$x_{ij}=\begin{cases}0 & \text{若城市}i\text{不到城市}j\\ 1 & \text{若城市}i\text{到城市}j,\text{且}i\text{在}j\text{前}\end{cases}$$

考虑每个城市后只有一个城市，则：

$$\sum_{\substack{j=1\\ j\neq i}}^{n} x_{ij}=1,\quad i=1,\cdots,n$$

考虑每个城市前只有一个城市，则：

$$\sum_{\substack{i=1\\ i\neq j}}^{n} x_{ij}=1,\quad j=1,\cdots,n\,;$$

两种方案

设有6个城市，下面矩阵代表了一种方案：

$$\begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\1&0&0&0&0&0\end{bmatrix}\qquad\qquad\begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\1&0&0&0&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&1&0&0\end{bmatrix}$$

合理方案

1-->2-->3-->4-->5-->6-->1

不合理方案

1-->2-->3--1　4-->5-->6-->4

但仅以上约束条件不能避免在一次遍历中产生多于一个互不连通回路。

为此我们引入额外变量 u_i $(i=1,\cdots,n)$，附加以下充分约束条件：

$$u_i-u_j+nx_{ij}\le n-1,\quad 1<i\neq j\le n\,;$$

该约束的解释：

如 i 与 j 不会构成回路，若构成回路，有：

$x_{ij}=1$，$x_{ji}=1$，则：

$u_i-u_j\le -1$，$u_j-u_i\le -1$，从而有：

$0\le -2$，导致矛盾。

如 i，j 与 k 不会构成回路，若构成回路，有：

$x_{ij}=1$，$x_{jk}=1$，$x_{ki}=1$ 则：

$u_i-u_j\le -1$，$u_j-u_k\le -1$，$u_k-u_i\le -1$ 从而有：

$0\le -3$，导致矛盾。

其它情况以此类推。

于是我们可以得到如下的模型：

$$\min Z=\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}x_{ij}$$

$$s.t.\begin{cases}\sum\limits_{\substack{i=1\\ i\neq j}}^{n}x_{ij}=1, & j=1,\cdots,n\\ \sum\limits_{\substack{j=1\\ j\neq i}}^{n}x_{ij}=1, & i=1,\cdots,n\\ u_i-u_j+nx_{ij}\le n-1, & 1<i\neq j\le n\\ x_{ij}=0\text{或}1, & i,j=1,\cdots,n\\ u_i\text{为实数}, & i=1,\cdots,n\end{cases} \qquad (1)$$

该模型的 Lingo 程序

```
!TSP quesion;
MODEL:
SETS:
city/1..10/:u;
link(city,city):d,x;
ENDSETS
DATA:
d= 0   7    4  5  8    6    12  13  11  18
   7   0    3 10  9    14   5   14  17  17
   4    3   0  5  9    10   21  8   27  12
   5   10 5     0  14  9    10  9   23  16
   8   9    9 14  0    7    8   7   20  19
   6   14 10    9  7   0    13  5   25  13
  12    5 21  10  8    13   0   23  21  18
  13   14 8     9  7   5    23  0   18  12
  11   17 27    23 20  25   21  18  0   16
  18   17 12    16 19  13   18  12  16  0;
 @text()=@writefor(link(i,j)|x(i,j)#GT#0:' x(',i,',',j,')=',x(i,j));
ENDDATA
  MIN=@SUM(link:d*x);
```

```
  @for(city(j):@sum(city(i)|j#ne#i:x(i,j))=1); !城市j前有一个城市相连;
  @for(city(i):@sum(city(j)|j#ne#i:x(i,j))=1); !城市i后有一个城市相连;
  @for(link(i,j)|i#NE#j#and#i#gt#1:u(i)-u(j)+10*x(i,j)<=9);
@FOR(link:@BIN(x));
End
```

得到的结果如下：
X(3,2)=1,X(4,1)=1,X(4,3)=1,X(6,5)=1,X(7,2)=1,X(7,5)=1,X(8,6)=1,X(9,1)=1,X(10,8)=1,X(10,9)=1。其它全为0。

其最短路线为 1—4—3—2—7—5—6—8—10—9—1，最短距离为 77 公里。

问题 2. 比赛项目排序问题

全民健身计划是 1995 年在国务院领导下，由国家体委会同有关部门、各群众组织和社会团体共同推行的一项依托社会、全民参与的体育健身计划，是与实现社会主义现代化目标相配套的社会系统工程和跨世纪的发展战略规划。现在，以全民健身为主要内容的群众性体育活动蓬勃开展，举国上下形成了全民健身的热潮，人民群众健康水平不断提高，同时也扩大了竞技体育的社会影响，提高了竞技体育水平。现在各级、各类、各种运动比赛比比皆是，这不但提高了全民的身体素质，而且使一批运动员脱颖而出，成为运动健将，为国家争得了荣誉。

在各种运动比赛中，为了使比赛公平、公正、合理的举行，一个基本要求是：在比赛项目排序过程中，尽可能使每个运动员不连续参加两项比赛，以便运动员恢复体力，发挥正常水平。

1．**表 2** 是某个小型运动会的比赛报名表。有 14 个比赛项目，40 名运动员参加比赛。表中第 1 行表示 14 个比赛项目，第 1 列表示 40 名运动员，表中“＃”号位置表示运动员参加此项比赛。建立此问题的数学模型，并且合理安排比赛项目顺序，使连续参加两项比赛的运动员人次尽可能的少；

2．文件“运动员报名表”中给出了某个运动比赛的报名情况。共有 61 个比赛项目，1050 人参加比赛。请给出算法及其框图，同时给出合理的比赛项目排序表，使连续参加两项比赛的运动员人次尽可能的少；

3．说明上述算法的合理性；

4．对“问题 2”的比赛排序结果，给出解决“运动员连续参加比赛”问题的建议及方案。

表 2　某小型运动会的比赛报名表

项目 运动员	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		#	#						#				#	
2								#			#	#		
3		#		#						#				
4			#					#				#		
5											#		#	#
6					#	#								
7												#	#	
8										#				#
9		#		#						#	#			
10	#	#		#			#							
11		#		#									#	#
12								#		#				
13					#					#				#
14			#	#				#						
15			#					#				#		
16									#		#	#		
17						#								#
18							#					#		
19			#							#				
20	#			#										
21									#					#
22		#			#									
23							#					#		
24							#	#					#	#
25	#	#								#				
26					#									#
27						#					#			
28		#						#						
29	#										#	#		
30				#	#									
31						#		#				#		
32							#			#				
33				#		#								
34	#		#										#	#
35					#	#						#		
36				#			#							
37	#								#	#				
38						#		#		#				#
39					#			#	#				#	
40						#	#		#				#	

问题 1 解答：

若项目 i 和项目 j 相邻，可以计算出同时参加这两个项目的人数，作为 i 和 j 的距离 d_{ij}。则问题转化为求项目 1 到项目 14 的一个排列，使相邻距离和最小。我们采用 TSP 问题求解。但由于开始项目和结束项目没有连接，可考虑引入虚拟项目 15，该虚拟项目与各个项目的距离都为 0。

距离矩阵 D 的求法：

该报名表用矩阵 $A_{40\times 14}$ 表示。

$$a_{ij}=\begin{cases}1 & \text{第}i\text{个人参加项目}j\\ 0 & \text{第}i\text{个人不参加项目}j\end{cases}$$

则 $d_{ij}=\sum_{k=1}^{40}a_{ki}.a_{kj} \qquad i\neq j,i,j,=1,2,\cdots,14$

$d_{ii}=0 \qquad i=1,2,\cdots,14$

另外 $d_{i,15}=0,d_{15,i}=0 \qquad i=1,2,\cdots,15$

由于问题 1 中 40 个运动员参加 14 个项目的比赛是 word 表，可将其拷贝到 Excel 表中，然后将#替换为 1，将空格替换为 0，形成 0-1 表，并拷贝到数据文件 table1.txt 中。问题 2 中 1050 个运动员参加的 61 个项目比赛的 Access 数据库中的表保存为 Excel 表，然后在表中将#替换为 1，将空格替换为 0，形成 0-1 表，并拷贝到数据文件 table2.txt 中。

在 Matlab 中编制如下程序 B2005.m 形成距离矩阵。

```
load table1.txt;
a=table1;

[m,n]=size(a);
d=zeros(n+1,n+1);  %定义距离矩阵;

for i=1:n
for j=1:n
  for k=1:m
    d(i,j)=d(i,j)+a(k,i)*a(k,j);  %计算不同项目之间距离
  end
end
end

for i=1:n+1
    d(i,i)=0;
end
```

```
 %输出文件
 fid=fopen('d:\lingo12\dat\ds1.txt','w');
for i=1:n+1
 for j=1:n+1
   fprintf(fid,'%1d ',d(i,j));
  end
 fprintf(fid,'\r\n');
end
 fclose(fid);
```

输出的距离矩阵 D 为 dis1.txt：

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

Lingo 程序 B2005.lg4：

```
!第一个问题的求解的程序：
!比赛项目排序问题;
model:
sets:
  item / 1.. 15/: u;
  link( item, item):dist,x;
endsets
   n = @size( item);
data:   !距离矩阵;
    dist=@file('d:\lingo12\dat\ds1.txt'); !文件路径;
      !输出为1的变量;
 @text()=@writefor(link(i,j)|x(i,j)#GT#0:' x(',i,',',j,')=',x(i,j));
enddata

  MIN=@SUM(link:dist*x);
  @for(item(j):@sum(item(i)|j#ne#i:x(i,j))=1); !点j前有一个点相连;
```

```
  @for(item(i):@sum(item(j)|j#ne#i:x(i,j))=1); !点i后前有一个点;
!保证不出现子圈;
  @for(link(i,j)|i#NE#j#and#i#gt#1:u(i)-u(j)+n*x(i,j)<=n-1);
@FOR(link:@BIN(x));!定义X为0-1变量;
end
```

其中数据文件dis1.txt在Matlab程序B2005.m中输出。

Lingo12求解结果为:
目标值z=2

```
x(1,8)=1   x(2,6)=1   x(3,11)=1  x(4,13)=1  x(5,1)=1
x(6,3)=1   x(7,5)=1   x(8,15)=1  x(9,4)=1   x(10,12)=1
x(11,7)=1  x(12,14)=1 x(13,10)=1 x(14,2)=1  x(15,9)=1
```

由于15是虚拟项，去掉后对应序列为

9-4-13-10-12-14-2-6-3-11-7-5-1-8
则项目排序如下，其中箭头上所示数字为连续参加相邻两项目的运动员数。

$$9 \xrightarrow{0} 4 \xrightarrow{1} 13 \xrightarrow{0} 10 \xrightarrow{0} 12 \xrightarrow{0} 14 \xrightarrow{1} 2$$

$$\xrightarrow{0} 6 \xrightarrow{0} 3 \xrightarrow{0} 11 \xrightarrow{0} 7 \xrightarrow{0} 5 \xrightarrow{0} 1 \xrightarrow{0} 8$$

即有两名运动员连续参加比赛。

问题2解答与问题1相同，只是项目变成61个，引入虚拟项目后变为62个，运动员为1050名。模型建立同问题1。在问题一中的Matlab程序中只需要将表table1.txt改为table2.txt，输出数据文件将dis1.txt改为dis2.txt就可以了。

在Lingo程序中将项目数由15修改为62，使用的数据文件由15改为62，同样可以运行，只是运行时间较长，本程序在Lingo12中大约运行6分钟左右。原始数据文件table2.txt和Matlab输出的距离矩阵dis2.txt，由于数据较大这里不列出，可参见附录。

Lingo程序

```
  !第二个问题的求解的程序:
!比赛项目排序问题;
model:
sets:
  item / 1.. 62/: u;
  link( item, item):dist,x;
endsets
   n = @size( item);
data:   !距离矩阵;
    dist=@file('d:\lingo12\dat\ds2.txt'); !文件路径;
     !输出为1的变量;
```

```
 @text()=@writefor(link(i,j)|x(i,j)#GT#0:' x(',i,',',j,')=',x(i,j));
enddata

  MIN=@SUM(link:dist*x);
  @for(item(j):@sum(item(i)|j#ne#i:x(i,j))=1); !点j前有一个点相连;
  @for(item(i):@sum(item(j)|j#ne#i:x(i,j))=1); !点i后前有一个点;
!保证不出现子圈;
  @for(link(i,j)|i#NE#j#and#i#gt#1:u(i)-u(j)+n*x(i,j)<=n-1);
@FOR(link:@BIN(x));!定义X为0-1变量;
end
```

Lingo12求解结果为:

目标值z=5

x(1,19)=1 x(2,44)=1   x(3,50)=1 x(4,25)=1 x(5,20)=1
x(6,15)=1 x(7,42)=1   x(8,59)=1 x(9,35)=1 x(10,3)=1
x(11,54)=1 x(12,21)=1 x(13,32)=1 x(14,41)=1 x(15,40)=1
x(16,57)=1 x(17,22)=1 x(18,9)=1   x(19,60)=1 x(20,6)=1
x(21,10)=1 x(22,37)=1 x(23,14)=1 x(24,51)=1 x(25,13)=1
x(26,27)=1 x(27,29)=1 x(28,17)=1 x(29,24)=1 x(30,58)=1
x(31,12)=1 x(32,56)=1 x(33,47)=1 x(34,23)=1 x(35,46)=1
x(36,45)=1 x(37,30)=1 x(38,49)=1 x(39,31)=1 x(40,48)=1
x(41,1)=1 x(42,52)=1 x(43,38)=1 x(44,4)=1 x(45,7)=1
x(46,62)=1 x(47,55)=1 x(48,34)=1 x(49,26)=1 x(50,36)=1
x(51,16)=1 x(52,18)=1 x(53,39)=1 x(54,43)=1 x(55,5)=1
x(56,11)=1 x(57,53)=1 x(58,61)=1 x(59,28)=1 x(60,8)=1
x(61,2)=1 x(62,33)=1

由于62是虚拟项，去掉后对应序列为

33—47—55—5—20—6—15—40—48—34—23—14-41—1—19—60—8—59—28—17—22—37—30—58—61—2—44—4—25—13—32—56—11—54—43—38—49—26—27—29—24—51—16—57—53—39—31—12—21—10—3—50—36—45—7—42—52—18—9—35—46

可以验证，其中 d(14,41)=1, d(51,16)=1, d(31,12)=1, d(10,3)=1, d(45,7)=1。其余相邻两个项目比赛没有两名运动连续参加比赛。即有 5 名运动员连续参加比赛。

另外，该问题解不唯一，单目标值都为5。

x(1,41)=1 x(2,61)=1 x(3,10)=1 x(4,44)=1 x(5,55)=1 x(6,20)=1 x(7,45)=1
x(8,60)=1 x(9,18)=1 x(10,21)=1 x(11,54)=1 x(12,31)=1 x(13,32)=1
x(14,23)=1 x(15,6)=1 x(16,57)=1 x(17,28)=1 x(18,52)=1 x(19,1)=1
x(20,5)=1 x(21,12)=1 x(22,17)=1 x(23,34)=1 x(24,51)=1 x(25,4)=1
x(26,27)=1 x(27,29)=1 x(28,59)=1 x(29,24)=1 x(30,37)=1 x(31,39)=1
x(32,56)=1 x(33,62)=1 x(34,48)=1 x(35,9)=1 x(36,50)=1 x(37,22)=1
x(38,49)=1 x(39,13)=1 x(40,15)=1 x(41,14)=1 x(42,7)=1 x(43,38)=1

x(44,2)=1 x(45,36)=1 x(46,35)=1 x(47,33)=1 x(48,40)=1 x(49,26)=1
x(50,3)=1 x(51,16)=1 x(52,42)=1 x(53,25)=1 x(54,43)=1 x(55,47)=1
x(56,11)=1 x(57,53)=1 x(58,30)=1 x(59,8)=1 x(60,19)=1 x(61,58)=1
x(62,46)=1

问题扩展：碎纸片复原问题

对于给定的来自同一页印刷文字文件的碎纸机破碎纸片（仅纵切），建立碎纸片拼接复原模型和算法。

我们依据像素特点抽取位于最左边和最右边的图片，定义了图片i的右边界与图片j的左边界的平均像素差作为i到j的距离，从而得到19×19的距离矩阵。

建立模型以所得序列横向相邻两图片的平均距离最小为目标，可采用 TSP 模型求解。