图论中TSP问题的LINGO求解与应用

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。该问题可简单描述为走遍n个城市的最短路。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。该问题在数学建模竞赛中也经常碰到。

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

两种方案比较

设有6个城市，下面矩阵代表了一种方案:

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

合理方案

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

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

不合理方案

1-->2-->3--1　4-->5-->6-->4

建立线性规划模型：

设城市之间距离用矩阵d表示，d_{ij}表示城市i与城市j距离。

设0--1矩阵X用来表示经过的各城市之间的路线。设

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

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

但该约束不能避免产生子圈。为此引入额外变量u_i，附加以下约束：

$$u_i - u_j + n x_{ij} \le n-1, \quad 1 < i \ne j \le n \text{；}$$

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 $0 \le -2$，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$0 \le -3$，导致矛盾。

其它情况以此类推。

总线性规划模型

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

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

LINGO程序：

```
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
 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后前有一个城市相连;
  @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
```

```
Solution Report - Lingo1

Global optimal solution found.
Objective value:                              77.00000
Objective bound:                              77.00000
Infeasibilities:                              0.000000
Extended solver steps:                               2
Total solver iterations:                          1247


x(1,9)=1 x(2,3)=1 x(3,4)=1 x(4,1)=1 x(5,7)=1 x(6,5)=1 x(7,2)=1 x(8,6)=1 x(9,10)=1 x(10,8)=1
```

图1　LINGO求解结果

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

问题2．比赛项目排序问题(2005年电工杯B题)

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。

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											#		#	#
33				#		#								
34	#		#										#	#
35					#	#						#		
36				#			#							
37	#								#	#				
38						#		#		#				#
39					#			#	#				#	
40						#	#		#				#	

解答：

若项目i和j相邻，计算同时参加这两个项目人数，作为i和j距离d(i,j)。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。

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。

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。这样让链和圈等价。

距离矩阵d的求法：

该报名表用矩阵 $A_{40\times 14}$ 表示。

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

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

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

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

Matlab实现程序：

```
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('dis1.txt','w');
for i=1:n+1
 for j=1:n+1
   fprintf(fid,'%1d ',d(i,j));
  end
 fprintf(fid,'\n');
end
 fclose(fid);
```

```
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
```

输出的距离矩阵d

比赛项目排序问题LINGO程序

```
model:
sets:
  item / 1.. 15/: u;
  link( item, item):dist,x;
endsets
   n = @size( item);
data:   !距离矩阵;
dist=@file(‘c:\lingo12\prg\dis1.txt’); !给指定距离矩阵文件路径;
@text()=@writefor(link(i,j)|x(i,j)#GT#0:' x(',i,',',j,')=',x(i,j));
!只输出为1的变量;
enddata
```

```
MIN=@SUM(link:dist*x);
@for(item(j):@sum(item(i)|j#ne#i:x(i,j))=1);
!点j前只一个点;
@for(item(i):@sum(item(j)|j#ne#i:x(i,j))=1); !点i后只有一个点;
@for(link(i,j)|i#NE#j#and#i#gt#1:u(i)-u(j)+n*x(i,j)<=n-1); !保证不出现子圈;
@FOR(link:@BIN(x));!定义X为0-1变量;
end
```

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，去掉后对应项目排序为

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

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

箭头上所示数字为连续参加相邻两项目的运动员数。

谢 谢！