R软件在最优化中的应用

魏太云

Email: weitaiyun@google.com

中南大学 数学院

2008 年 12 月

内容提要

1. 线性规划与整数规划

内容提要

内容提要

内容提要

线性规划与整数规划

线性规划与整数规划

模型

$$
\begin{aligned}
&\min \quad z = \mathbf{C}\mathbf{x} \\
&s.t. \begin{cases}
\mathbf{A}\mathbf{x} \leqslant \mathbf{b} \\
\mathbf{x} \geqslant \mathbf{0} \\
\mathbf{l} \leqslant \mathbf{X} \leqslant \mathbf{u} \\
\mathbf{x} \text{ 中的元素取整数、 0 - 1 整数或实数}
\end{cases}
\end{aligned}
\tag{1.1}
$$

线性规划与整数规划

模型

$$\min \quad z = \mathbf{Cx}$$
$$s.t.\begin{cases} \mathbf{Ax} \leqslant \mathbf{b} \\ \mathbf{x} \geqslant \mathbf{0} \\ \mathbf{l} \leqslant \mathbf{X} \leqslant \mathbf{u} \\ \mathbf{x} \text{ 中的元素取整数、 0 - 1 整数或实数} \end{cases} \tag{1.1}$$

- 模型适用于：

线性规划与整数规划

模型

$$
\begin{aligned}
&\min \quad z = \mathbf{Cx} \\
&s.t. \begin{cases} \mathbf{Ax} \leqslant \mathbf{b} \\ \mathbf{x} \geqslant \mathbf{0} \\ \mathbf{l} \leqslant \mathbf{X} \leqslant \mathbf{u} \\ \mathbf{x} \text{ 中的元素取整数、 0 - 1 整数或实数} \end{cases}
\end{aligned}
\tag{1.1}
$$

- 模型适用于：
  1. 线性规划问题

线性规划与整数规划

模型

$$
\begin{aligned}
&\min \quad z = \mathbf{Cx} \\
&s.t. \begin{cases}
\mathbf{Ax} \leqslant \mathbf{b} \\
\mathbf{x} \geqslant \mathbf{0} \\
\mathbf{l} \leqslant \mathbf{X} \leqslant \mathbf{u} \\
\mathbf{x}\ \text{中的元素取整数、0 - 1 整数或实数}
\end{cases}
\end{aligned}
\tag{1.1}
$$

- 模型适用于：
  1. 线性规划问题
  2. 整数规划问题

线性规划与整数规划

模型

$$\min \quad z = \mathbf{Cx}$$
$$s.t. \begin{cases} \mathbf{Ax} \leqslant \mathbf{b} \\ \mathbf{x} \geqslant \mathbf{0} \\ \mathbf{l} \leqslant \mathbf{X} \leqslant \mathbf{u} \\ \mathbf{x}\ \text{中的元素取整数、}\ 0\text{-}1\ \text{整数或实数} \end{cases} \tag{1.1}$$

- 模型适用于：
  1. 线性规划问题
  2. 整数规划问题
  3. 混合整数规划问题

线性规划与整数规划

模型

$$\min \quad z = \mathbf{Cx}$$
$$s.t. \begin{cases} \mathbf{Ax} \leqslant \mathbf{b} \\ \mathbf{x} \geqslant \mathbf{0} \\ \mathbf{l} \leqslant \mathbf{X} \leqslant \mathbf{u} \\ \mathbf{x} \text{ 中的元素取整数、 0-1 整数或实数} \end{cases} \tag{1.1}$$

- 模型适用于：
  1. 线性规划问题
  2. 整数规划问题
  3. 混合整数规划问题
- 线性规划、整数规划都是混合整数规划的特例。

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

求下列混合整数规划规划问题:

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

求下列混合整数规划规划问题：

$$
\max \quad z = 3x_1 + x_2 + 3x_3
$$
$$
s.t. \begin{cases} \begin{array}{rcrcrcr} -x_1 & + & 2x_2 & + & x_3 & \leqslant & 4 \\ & & 4x_2 & - & 3x_3 & \leqslant & 2 \\ x_1 & - & 3x_2 & + & 2x_3 & \leqslant & 3 \end{array} \\ x_1,\ x_3 \text{ 为正整数} \end{cases}
$$

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

求下列混合整数规划规划问题：

$$\max\ z = 3x_1 + x_2 + 3x_3$$
$$s.t. \begin{cases} -x_1 + 2x_2 + x_3 \leqslant 4 \\ 4x_2 - 3x_3 \leqslant 2 \\ x_1 - 3x_2 + 2x_3 \leqslant 3 \\ x_1,\ x_3 \text{ 为正整数} \end{cases}$$

代码如下：

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

求下列混合整数规划规划问题：

$$
\max \quad z = 3x_1 + x_2 + 3x_3
$$
$$
s.t. \begin{cases} -x_1 + 2x_2 + x_3 \leqslant 4 \\ \quad 4x_2 - 3x_3 \leqslant 2 \\ x_1 - 3x_2 + 2x_3 \leqslant 3 \\ x_1,\ x_3 \text{ 为正整数} \end{cases}
$$

代码如下：

```
> obj <- c(3, 1, 3)
> mat <- matrix(c(-1, 0, 1, 2, 4, -3, 1, -3, 2), nrow = 3)
> dir <- rep("<=", 3)
> rhs <- c(4, 2, 3)
> types <- c("I", "C", "I")          ##变量类型
> Rglpk_solve_LP(obj, mat, dir, rhs, types, max = TRUE)
```

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

- 输出结果为:

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

- 输出结果为：

```
$optimum
[1] 26.75

$solution
[1] 5.00 2.75 3.00

$status
[1] 0
```

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

- 输出结果为:

```
$optimum
[1] 26.75

$solution
[1] 5.00 2.75 3.00

$status
[1] 0
```

1. $optimum 为目标函数最大值

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

- 输出结果为：

```
$optimum
[1] 26.75

$solution
[1] 5.00 2.75 3.00

$status
[1] 0
```

1. $optimum 为目标函数最大值
2. $solution 为最优解

Rglpk_solve_LP(Rglpk) 函数求解混合整数规划

- 输出结果为:

```
$optimum
[1] 26.75

$solution
[1] 5.00 2.75 3.00

$status
[1] 0
```

1. $optimum 为目标函数最大值
2. $solution 为最优解
3. $status 为逻辑变量，为 0 时表示求解成功

运输问题和指派问题

运输问题和指派问题

- 运输问题：特殊的线性规划问题

运输问题和指派问题

- 运输问题：特殊的线性规划问题
- 指派问题： 0-1 整数规划问题

运输问题和指派问题

- 运输问题：特殊的线性规划问题
- 指派问题：0 - 1 整数规划问题

  用 lpSolve 包可以求解运输问题和指派问题
- 运输问题：函数 lp.transport()

运输问题和指派问题

- 运输问题：特殊的线性规划问题
- 指派问题： 0 - 1 整数规划问题

  用 lpSolve 包可以求解运输问题和指派问题
- 运输问题：函数 lp.transport()

```
lp.transport(cost.mat,direction="min",row.signs,row.rhs,col.signs,
             col.rhs,presolve=0,compute.sens=0,integers=1:(nc*nr))
```

- 求解指派问题：函数 lp.assign()

运输问题和指派问题

- 运输问题：特殊的线性规划问题
- 指派问题： 0 - 1 整数规划问题

  用 lpSolve 包可以求解运输问题和指派问题
- 运输问题：函数 lp.transport()

```
lp.transport(cost.mat,direction="min",row.signs,row.rhs,col.signs,
             col.rhs,presolve=0,compute.sens=0,integers=1:(nc*nr))
```

- 求解指派问题：函数 lp.assign()

```
lp.assign (cost.mat,direction="min",presolve=0,compute.sens=0)
```

目标规划

模型

$$
\begin{aligned}
&\min P_l\Big(\sum_{k=1}^{K}(W_{lk}^{-}d_k^{-}+W_{lk}^{+}d_k^{+})\Big) \quad l=1,2,\ldots,L\\
&\text{s.t.}\begin{cases}
\sum_{j=1}^{n} c_{kj}x_j + d_k^{-} - d_k^{+} = g_k & k=1,2,\ldots,K\\
\sum_{j=1}^{n} a_{ij}x_j \leqslant (\text{或}=,\text{或}\geqslant) b_i & i=1,2,\ldots,m\\
x_j \geqslant 0 & j=1,2,\ldots,n\\
d_k^{-},\ d_k^{+} \geqslant 0 & k=1,2,\ldots,K
\end{cases}
\end{aligned}
\tag{2.1}
$$

目标规划

模型

$$
\begin{array}{ll}
\min P_l(\sum_{k=1}^{K}(W_{lk}^{-}d_k^{-}+W_{lk}^{+}d_k^{+})) & l=1,2,\ldots,L \\
\text{s.t.}\left\{\begin{array}{ll}
\sum_{j=1}^{n} c_{kj}x_j+d_k^{-}-d_k^{+}=g_k & k=1,2,\ldots,K \\
\sum_{j=1}^{n} a_{ij}x_j \leqslant (\text{或}=,\text{或}\geqslant) b_i & i=1,2,\ldots,m \\
x_j \geqslant 0 & j=1,2,\ldots,n \\
d_k^{-},\ d_k^{+} \geqslant 0 & k=1,2,\ldots,K
\end{array}\right.
\end{array}
\tag{2.1}
$$

1. 第一行约束为目标约束条件

目标规划

模型

$$
\begin{aligned}
&\min P_l\left(\sum_{k=1}^{K}(W_{lk}^{-}d_k^{-}+W_{lk}^{+}d_k^{+})\right) \quad l=1,2,\ldots,L\\
&\text{s.t.}\begin{cases}
\sum_{j=1}^{n} c_{kj}x_j + d_k^{-} - d_k^{+} = g_k & k=1,2,\ldots,K\\
\sum_{j=1}^{n} a_{ij}x_j \leqslant (\text{或} =, \text{或} \geqslant) b_i & i=1,2,\ldots,m\\
x_j \geqslant 0 & j=1,2,\ldots,n\\
d_k^{-},\ d_k^{+} \geqslant 0 & k=1,2,\ldots,K
\end{cases}
\end{aligned}
\tag{2.1}
$$

1. 第一行约束为目标约束条件
2. 第二行约束为绝对约束条件

llgp(goalprog) 函数求解目标规划

在模型 (2.1) 有解的情况下，可以将其化为：

模型（llgp()函数要求的格式)

$$\min \mathbf{P}(\mathbf{W}^-\mathbf{d}^- + \mathbf{W}^+\mathbf{d}^+)$$
$$s.t. \begin{cases} \mathbf{A}\mathbf{x} + \mathbf{d}^- - \mathbf{d}^+ = \mathbf{g} \\ \mathbf{d}^-,\ \mathbf{d}^+ \geqslant \mathbf{0} \\ \mathbf{x} \geqslant \mathbf{0} \end{cases} \tag{2.2}$$

llgp(goalprog) 函数求解目标规划

在模型 (2.1) 有解的情况下，可以将其化为：

模型 (llgp()函数要求的格式)

$$\min \mathbf{P}(\mathbf{W}^-\mathbf{d}^- + \mathbf{W}^+\mathbf{d}^+)$$
$$s.t. \begin{cases} \mathbf{A}\mathbf{x} + \mathbf{d}^- - \mathbf{d}^+ = \mathbf{g} \\ \mathbf{d}^-, \ \mathbf{d}^+ \geqslant \mathbf{0} \\ \mathbf{x} \geqslant \mathbf{0} \end{cases} \tag{2.2}$$

- goalprog 包专门求解目标规划问题，核心函数为llgp()

llgp(goalprog) 函数求解目标规划

在模型 (2.1) 有解的情况下，可以将其化为：

模型（llgp()函数要求的格式)

$$\min \mathbf{P}(\mathbf{W}^-\mathbf{d}^- + \mathbf{W}^+\mathbf{d}^+)$$
$$s.t. \begin{cases} \mathbf{A}\mathbf{x} + \mathbf{d}^- - \mathbf{d}^+ = \mathbf{g} \\ \mathbf{d}^-, \ \mathbf{d}^+ \geqslant \mathbf{0} \\ \mathbf{x} \geqslant \mathbf{0} \end{cases} \tag{2.2}$$

- goalprog 包专门求解目标规划问题，核心函数为llgp()

```
llgp(coefficients, targets, achievements,...)
```

llgp(goalprog) 函数求解目标规划

llgp(goalprog) 函数求解目标规划

- 求下列目标规划问题:

$$\min \{P_1(2d_1^+ + 3d_2^+),\ P_2d_3^-,\ P_3d_4^+\}$$

$$\text{s.t.} \begin{cases} x_1 + x_2 + d_1^- - d_1^+ = 10 \\ x_1 + d_2^- - d_2^+ = 4 \\ 5x_1 + 3x_2 + d_3^- - d_3^+ = 56 \\ x_1 + x_2 + d_4^- - d_4^+ = 12 \\ x_1,\ x_2,\ d_{1-4}^-,\ d_{1-4}^+ \geqslant 0 \end{cases}$$

llgp(goalprog) 函数求解目标规划

- 求下列目标规划问题：

$$\min\ \{P_1(2d_1^+ + 3d_2^+),\ P_2d_3^-,\ P_3d_4^+\}$$

$$\text{s.t.}\ \begin{cases} x_1 + x_2 + d_1^- - d_1^+ = 10 \\ x_1 + d_2^- - d_2^+ = 4 \\ 5x_1 + 3x_2 + d_3^- - d_3^+ = 56 \\ x_1 + x_2 + d_4^- - d_4^+ = 12 \\ x_1,\ x_2,\ d_{1-4}^-,\ d_{1-4}^+ \geqslant 0 \end{cases}$$

- 代码如下：

```
> coefficients = matrix(c(1,1,5,1,1,0,3,1), 4)
> targets = c(10,4,56,12)
> achievements = data.frame(objective=1:4,
+ priority=c(1,1,2,3), p=c(2,3,0,1), n=c(0,0,1,0))
> soln = llgp(coefficients, targets, achievements)
```

部分参数及结果

```
> achievements                    #输出偏差变量相关信息数据框
  objective priority p n
1         1        1 2 0
2         2        1 3 0
3         3        2 0 1
4         4        3 1 0
> soln$converged                  #若为TRUE，则表示求得最优解
[1] TRUE
> soln$out                        #得到的解
Decision variables
                 X
X1    4.000000e+00
X2    6.000000e+00
```

非线性规划

模型

$$
\begin{aligned}
&\min \quad z = f(\mathbf{x}) \\
&s.t. \begin{cases} \mathbf{x_l} & \leqslant & \mathbf{x} & \leqslant & \mathbf{x_u} \\ \mathbf{b_l} & \leqslant & \mathbf{Ax} & \leqslant & \mathbf{b_u} \\ \mathbf{c_l} & \leqslant & \mathbf{c}(\mathbf{x}) & \leqslant & \mathbf{c_u} \end{cases}
\end{aligned}
\tag{3.1}
$$

非线性规划

模型

$$
\begin{aligned}
&\min \quad z = f(\mathbf{x}) \\
&s.t. \begin{cases} \mathbf{x_l} & \leqslant & \mathbf{x} & \leqslant & \mathbf{x_u} \\ \mathbf{b_l} & \leqslant & \mathbf{Ax} & \leqslant & \mathbf{b_u} \\ \mathbf{c_l} & \leqslant & \mathbf{c}(\mathbf{x}) & \leqslant & \mathbf{c_u} \end{cases}
\end{aligned}
\tag{3.1}
$$

三个约束条件中：

非线性规划

模型

$$
\begin{aligned}
&\min \quad z = f(\mathbf{x}) \\
&s.t. \begin{cases} \mathbf{x_l} \leqslant \mathbf{x} \leqslant \mathbf{x_u} \\ \mathbf{b_l} \leqslant \mathbf{Ax} \leqslant \mathbf{b_u} \\ \mathbf{c_l} \leqslant \mathbf{c(x)} \leqslant \mathbf{c_u} \end{cases}
\end{aligned} \tag{3.1}
$$

三个约束条件中：

- 第一个为定义域约束

非线性规划

模型

$$\min \quad z = f(\mathbf{x})$$
$$s.t. \begin{cases} \mathbf{x_l} \leqslant \mathbf{x} \leqslant \mathbf{x_u} \\ \mathbf{b_l} \leqslant \mathbf{Ax} \leqslant \mathbf{b_u} \\ \mathbf{c_l} \leqslant \mathbf{c}(\mathbf{x}) \leqslant \mathbf{c_u} \end{cases} \tag{3.1}$$

三个约束条件中：

- 第一个为定义域约束
- 第二个为线性约束

非线性规划

模型

$$
\begin{aligned}
&\min \quad z = f(\mathbf{x}) \\
&s.t. \begin{cases} \mathbf{x_l} & \leqslant & \mathbf{x} & \leqslant & \mathbf{x_u} \\ \mathbf{b_l} & \leqslant & \mathbf{Ax} & \leqslant & \mathbf{b_u} \\ \mathbf{c_l} & \leqslant & \mathbf{c}(\mathbf{x}) & \leqslant & \mathbf{c_u} \end{cases}
\end{aligned}
\tag{3.1}
$$

三个约束条件中：

- 第一个为定义域约束
- 第二个为线性约束
- 第三个为非线性约束

非线性规划

模型

$$\min \quad z = f(\mathbf{x})$$
$$s.t. \begin{cases} \mathbf{x_l} & \leqslant & \mathbf{x} & \leqslant & \mathbf{x_u} \\ \mathbf{b_l} & \leqslant & \mathbf{Ax} & \leqslant & \mathbf{b_u} \\ \mathbf{c_l} & \leqslant & \mathbf{c(x)} & \leqslant & \mathbf{c_u} \end{cases} \tag{3.1}$$

三个约束条件中：

- 第一个为定义域约束
- 第二个为线性约束
- 第三个为非线性约束

当约束条件和目标函数为光滑时称之为光滑的非线性规划

donlp2(Rdonlp2) 函数求解光滑的非线性规划

```
donlp2(par, fn,
       par.upper=rep(+Inf, length(par)),
       par.lower=rep(-Inf, length(par)),

       A = NULL,
       lin.upper=rep(+Inf, length(par)),
       lin.lower=rep(-Inf, length(par)),

       nlin = list(),
       nlin.upper=rep(+Inf, length(nlin)),
       nlin.lower=rep(-Inf, length(nlin)),

       control=donlp2.control(),
       control.fun=function(lst)return(TRUE),
       env=.GlobalEnv,
       name="Rdonlp2")
```

求下列有约束的非线性规划问题:

$$\min \quad z = x^2 \sin y + y^2 \cos x$$

$$s.t. \begin{cases} -100 < x < 100 \\ -100 < y < 100 \\ 1 \leqslant 3x - y \leqslant 3 \\ x + y \geqslant 2 \\ \sin x \cos y \leqslant 0.6 \\ xy = 2 \end{cases}$$

代码

```
p = c(10,10)                                      #迭代初始值
par.l= c(-100,-100); par.u = c(100,100)     #自变量定义域约束
fn = function(x){
  x[1]^2*sin(x[2])+x[2]^2*cos(x[1])
}                                                 #目标函数
A = matrix(c(1,1,3,-1),2,byrow=TRUE)
lin.l = c(2,1);lin.u = c(+Inf,3)               #线性约束
nlcon1 = function(x){
  x[1]*x[2]
}
nlcon2 = function(x){
  sin(x[1])*cos(x[2])
}
nlin.l = c(2,-Inf)
nlin.u = c(2,0.6)                                  #非线性约束
ret = donlp2(p, fn, par.u=par.u, par.l=par.l,A,lin.l=lin.l,lin.u=lin.u,
             nlin=list(nlcon1,nlcon2), nlin.u=nlin.u, nlin.l=nlin.l)
```

用启发式算法求解一般的非线性规划

用启发式算法求解一般的非线性规划

- 遗传算法

用启发式算法求解一般的非线性规划

- 遗传算法
- 模拟退火算法

用启发式算法求解一般的非线性规划

- 遗传算法
  1. gafit 包：gafit() 函数
  2. genalg 包：rbga() 函数
  3. rgenoud 包：rgenoud() 函数
- 模拟退火算法

用启发式算法求解一般的非线性规划

- 遗传算法
  1. gafit 包：gafit() 函数
  2. genalg 包：rbga() 函数
  3. rgenoud 包：rgenoud() 函数
- 模拟退火算法
  1. stats 包：optim() 函数

igraph 包在图与网络中的应用

igraph 包在图与网络中的应用

图与网络中的几个经典问题:

igraph 包在图与网络中的应用

图与网络中的几个经典问题:

- 最大流问题
- 最小生成树问题
- 最短路问题

igraph 包在图与网络中的应用

图与网络中的几个经典问题:

- 最大流问题

  `graph.maxflow(graph, source, target, capacity=NULL)`

- 最小生成树问题

  `minimum.spanning.tree(graph, weights=NULL, algorithm=NULL, ...)`

- 最短路问题

  `shortest.paths(graph,v=V(graph),mode=c("all","out","in"),weights)`

igraph 包在图与网络中的应用

图与网络中的几个经典问题:

- 最大流问题

  `graph.maxflow(graph, source, target, capacity=NULL)`

- 最小生成树问题

  `minimum.spanning.tree(graph, weights=NULL, algorithm=NULL, ...)`

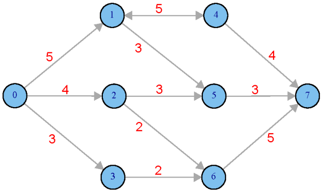
- 最短路问题

  `shortest.paths(graph,v=V(graph),mode=c("all","out","in"),weights)`

igraph 包非常强大，可以快速便捷地创建、绘制和分析无向图及有向图，图的顶点和边允许百万以上!

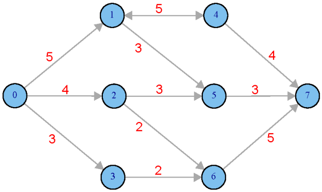
igraph 包在图与网络中的应用

下图是个有向图，方向如图中箭头所示，边上的数字为其权重:



igraph 包在图与网络中的应用

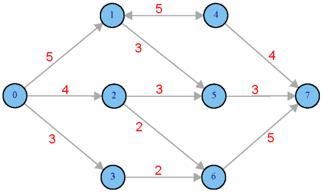
下图是个有向图，方向如图中箭头所示，边上的数字为其权重:



试求下列问题:

igraph 包在图与网络中的应用

下图是个有向图，方向如图中箭头所示，边上的数字为其权重:



试求下列问题：

1. 从顶点 0 到顶点 7 的最大流量 (此时图中各条边上的数字代表容量限制)；
2. 该连通图的最小生成树；
3. 该图中任意两顶点之间的最短路程 (考虑方向)。

代码

```
> e = matrix(nc = 3,byrow = TRUE,c(0,1,5, 0,2,4, 0,3,3, 1,5,3, 1,4,5,
+     2,5,3, 2,6,2, 3,6,2, 4,1,5, 4,7,4, 5,7,3, 6,7,5))#边的权矩阵
> g = add.edges(graph.empty(8),t(e[,1:2]), weight = e[,3]) #构造图
> tkplot(g)                                    #绘制网络图
> graph.maxflow(g, 0,7, capacity = E(g)$weight) #最大流
> mst = minimum.spanning.tree(g)               #最小生成树
> tkplot(mst)                                  #绘制最小生成树
> tree_min = sum(E(mst)$weight)                #计算并输出最小生成树的权
> shortest.paths(g, mode = "out")              #最短路矩阵
```

结果

结果

1. 最大流

```
> graph.maxflow(g, 0,7, capacity = E(g)$weight) #最大流
[1] 11
```

结果

1. 最大流

```
> graph.maxflow(g, 0,7, capacity = E(g)$weight) #最大流
[1] 11
```

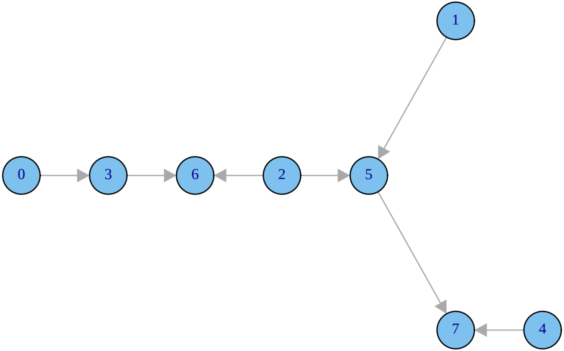
2. 最短路矩阵：

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]
[1,]	0	5	4	3	10	7	5	10
[2,]	Inf	0	Inf	Inf	5	3	Inf	6
[3,]	Inf	Inf	0	Inf	Inf	3	2	6
[4,]	Inf	Inf	Inf	0	Inf	Inf	2	7
[5,]	Inf	5	Inf	Inf	0	8	Inf	4
[6,]	Inf	Inf	Inf	Inf	Inf	0	Inf	3
[7,]	Inf	Inf	Inf	Inf	Inf	Inf	0	5
[8,]	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0

结果

- 最小生成树：

```
> sum(E(mst)$weight) #计算并输出最小生成树的权
[1] 20
```



旅行商问题

旅行商问题是图论和组合优化中的经典问题，属于 NP 难题。R 中，TSP 包专门求解旅行商问题，核心函数为 solve_TSP():

旅行商问题

旅行商问题是图论和组合优化中的经典问题，属于 NP 难题。R 中，TSP 包专门求解旅行商问题，核心函数为 solve_TSP():

```
solve_TSP(x, method, control)
```

旅行商问题

旅行商问题是图论和组合优化中的经典问题，属于 NP 难题。R 中，TSP 包专门求解旅行商问题，核心函数为 solve_TSP():

```
solve_TSP(x, method, control)
```

走遍中国问题:

旅行商问题

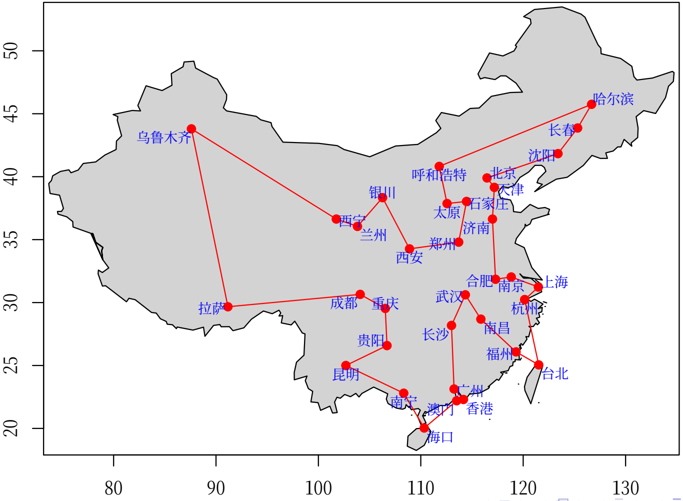
旅行商问题是图论和组合优化中的经典问题，属于 NP 难题。R 中，TSP 包专门求解旅行商问题，核心函数为 solve_TSP():

```
solve_TSP(x, method, control)
```

走遍中国问题:
你我周游全国，从北京出发，要遍游我国 34 个省级行政中心，最后要回到北京。假设各城市之间的路程可以视为它们在地球球面上的最短距离，请设计一条路线使得总行程最短。

走遍中国线路

走遍中国线路


50
45
40
35
30
25
20
80
90
100
110
120
130
哈尔滨
长春
沈阳
乌鲁木齐
呼和浩特
北京
天津
银川
石家庄
太原
西宁
兰州
济南
郑州
西安
合肥
南京
上海
武汉
杭州
拉萨
成都
重庆
长沙
南昌
贵阳
福州
台北
昆明
广州
南宁
澳门
香港
海口

结束语

Thank you!

Email: weitaiyun@gmail.com