第 4 章　图与网络模型及方法习题解答

4.1　解　求解的 Matlab 程序如下

```
clc, clear
a=zeros(6); %邻接矩阵初始化
a(1,[2:6])=[56 35   21   51   60]; %输入邻接矩阵的上三角元素
a(2,[3:6])=[21 57   78   70];
a(3,[4:6])=[36 68   68];
a(4,[5 6])=[51 61]; a(5,6)=13;
a=a'; a=sparse(a); %变换成下三角矩阵，并转化成工具箱所需要的稀疏矩阵
[ST,pred] = graphminspantree(a,'method','Kruskal') %调用工具箱求最小生成树
view(biograph(ST,[],'ShowArrows','off','ShowWeights','on'))
```

求得的最小生成树见图 4.1。

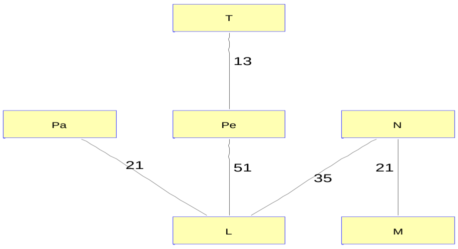

图 4. 1　最小生成树

4.2　解　记 v_i（$i=1,2,3,4$）表示第 i 年年初的时刻，v_5 表示第 4 年末的时刻，构造赋权图 $G=(V,A,W)$，其中 $V=\{v_1,\cdots,v_5\}$，A 为弧的集合，邻接矩阵 $W=(w_{ij})_{5\times5}$，这里 w_{ij} 为 v_i 到 v_j 的费用，例如，w_{12} 为第 1 年初到第 2 年初的费用，等于购置费用加维修费用减去机器处理价，$w_{12}=2.5+0.3-2.0=0.8$，可以计算得到

$$W=\begin{bmatrix} 0 & 0.8 & 2 & 3.8 & 6 \\ \infty & 0 & 0.9 & 2.1 & 3.9 \\ \infty & \infty & 0 & 1.1 & 2.3 \\ \infty & \infty & \infty & 0 & 1.4 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

4 年内用于更换、购买及运行维修总费用最省的问题，归结为求图 G 中从 v_1 到 v_5 的费用最短路，可以使用 Dijkstra 标号算法求解。

求解的 Matlab 程序如下

```
clc, clear
a=zeros(5); %邻接矩阵初始化
a(1,[2:5])=[0.8 2 3.8 6]; %输入邻接矩阵，注意这里实际上为有向图
a(2,[3:5])=[0.9 2.1 3.9];
a(3,[4,5])=[1.1 2.3]; a(4,5)=1.4;
b=sparse(a); %有向图直接变成稀疏矩阵就可以了
[dist,path]=graphshortestpath(b,1,5,'Directed',1) %调用工具箱求最短路
```

求得的最优更新策略为第 2 年初和第 3 年初都换一台新机器，总费用为 4 万元。

4.3　解　应用最大流算法必须是单源和单汇的网络。构造一个虚拟的源点 v_s，由于 A,B,C 的可供量分别为 20,20,100，则弧 v_sA，v_sB，v_sC 上的容量分别为 20,20,100，构造一个虚拟的汇点 v_t，由于市场 1,2,3,4 的需求量分别为 20,20,60,20，市场 1,2,3,4 分别记为 D,F,H,I，则弧 Dv_t，Fv_t，Hv_t，Iv_t 的容量分别为 20,20,60,20。构造赋权有向图 (V,E,W)，其中 V 为顶点集合，E 为弧的集合，W 为各个弧上的容量所构成的权重矩阵，具体计算时，把顶点 v_s,A,B,C,D,F,H,I,v_t 分别编号为 $1,2,\cdots,9$，从仓库到市场的最大流问题归结为求从 v_s 到 v_t 的最大流，可以使用 Ford－Fulkerson 算法求最大流。

使用 Matlab 软件求解最大流的程序如下

```
clc, clear
a=zeros(9); %容量矩阵初始化
a(1,[2:4])=[20,20,100]; %输入各弧上的容量
a(2,[5 6 8])=[30,10,40];
a(3,[7,8])=[10,50];
a(4,[5:8])=[20,10,40,5];
a([5:8],9)=[20,20,60,20];
a=sparse(a); %构造工具箱需要的稀疏矩阵
[b,c]=graphmaxflow(a,1,9) %调用工具箱求最大流的命令
```

求得从仓库运往市场的最大流量为 110 单位，其中市场 3 只能满足 50 单位，差 10 单位。

4.4　解　将五个人与五个外语语种分别用点表示，把各个人与懂得的外语语种之间用弧相连。为了求单源和单汇网络的最大流，再加一个虚拟的单源 v_s，v_s 与五个人之间各有一条弧，再加一个虚拟的单汇 v_t，在五个外语语种和 v_t 之间各有一条弧。规定每条弧的容量为 1，求出上述网络的最大流数字即为最多能得到招聘的人数。计算时把源点 v_s、甲乙丙丁戊五个人、俄英日德法五个外语语种和汇点 v_t 分别编号为 $1,2,\cdots,12$。

计算的 Matlab 程序如下

```
clc, clear
a=zeros(12); %容量矩阵初始化
a(1,[2:6])=1; %输入各弧上的容量,源点至五个人
a(2,[8,9])=1; %甲懂英，日
a(3,[7,8,10])=1; %乙懂俄，英，德
a(4,[8,9])=1; %丙懂英，日
a(5,[8,9])=1; %丁懂英，日
a(6,[10,11])=1; %戊懂德，法
a([7:11],12)=1; %五个外语语种到汇点
a=sparse(a); %化成稀疏矩阵
[b,c]=graphmaxflow(a,1,12) %调用工具箱求最大流的命令
```

求得只有 4 个人得到招聘，乙—俄，丙—日，丁—英，戊—德，甲未能得到应聘。

4.5 解 为了使用最大流算法，必须构造单源单汇的网络，加一个虚拟的源点 v_s，一个虚拟的汇点 v_t，得到的网络图如图 4.2 所示，其中弧旁的第一个数字为单位流的费用，第二个数字表示容量。

首先求出最大流的流量为 15。可以建立线性规划模型求解，求解的 Lingo 程序如下

```
model:
sets:
nodes/s,a,b,1,2,3,t/;
arcs(nodes,nodes)/s a,s b,a 1,a 2,a 3,b 1,b 2,b 3,1 t,2 t,3 t/:c,f;
endsets
data:
c=8 7 8 8 8 7 7 7 4 5 6;
enddata
n=@size(nodes); !顶点的个数;
max=flow;
@for(nodes(i)|i #ne#1 #and# i #ne# n:
        @sum(arcs(i,j):f(i,j))=@sum(arcs(j,i):f(j,i)));
@sum(arcs(i,j)|i #eq# 1:f(i,j))=flow;
@sum(arcs(i,j)|j #eq# n:f(i,j))=flow;
@for(arcs:@bnd(0,f,c));
end
```

再建立求最小费用的线性规划模型，求得最小费用为240。求最小费用的Lingo程序如下

```
model:
sets:
nodes/s,a,b,1,2,3,t/:d;
arcs(nodes,nodes)/s a,s b,a 1,a 2,a 3,b 1,b 2,b 3,1 t,2 t,3 t/:b,c,f;
endsets
data:
b=0 0 20 24 5 30 22 20 0 0 0;
c=8 7 8 8 8 7 7 7 4 5 6;
d=15 0 0 0 0 0 -15; !最大流为15;
enddata
n=@size(nodes); !顶点的个数;
min=@sum(arcs:b*f);
@for(nodes(i):@sum(arcs(i,j):f(i,j))-@sum(arcs(j,i):f(j,i))=d(i));
@for(arcs:@bnd(0,f,c));
end
```

4.6 解 （1）求最大流的 Matlab 程序如下

```
clc, clear
a=zeros(5); %容量矩阵初始化
a(1,[2 3])=[10 8]; %输入各弧上的容量
a(2,[4 5])=[2 7];
a(3,[2 4])=[5,10];
a(4,5)=4;
a=sparse(a); %构造工具箱需要的稀疏矩阵
[b,c]=graphmaxflow(a,1,5) %调用工具箱求最大流的命令
```

求得的最大流量为 11。

（2）求最大流的最小费用的 Lingo 程序如下

```
model:
sets:
nodes/s,1,2,3,t/:d;
arcs(nodes,nodes)/s 1,s 2,1 3,1 t,2 1,2 3,3 t/:b,c,f;
endsets
data:
b=4 1 6 1 2 3 2;
c=10 8 2 7 5 10 4;
d=11 0 0 0 -11; !最大流为11;
enddata
n=@size(nodes); !顶点的个数;
min=@sum(arcs:b*f);
@for(nodes(i):@sum(arcs(i,j):f(i,j))-@sum(arcs(j,i):f(j,i))=d(i));
@for(arcs:@bnd(0,f,c));
end
```

求得的最小费用为55。

4.7　解　（1）产品的计划网络图如图 4.18 所示

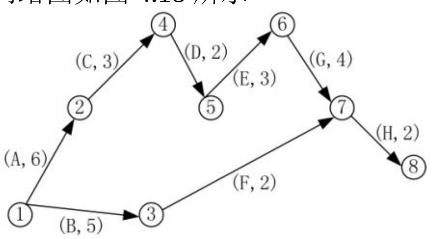

图 4.18　产品的计划网络图

（2）分别用 x_i, z_i 表示第 i（$i=1,\cdots,8$）个事件的最早时间和最迟时间，t_{ij} 表示作业 (i, j) 的计划时间，$es_{ij}, ls_{ij}, ef_{ij}, lf_{ij}$ 分别表示作业 (i, j) 的最早开工时间，最迟开工时间，最早完工时间，最晚完工时间。对应作业的最早开工时间与最迟开工时间相同，就得到项目的关键路径。

为了求事件的最早开工时间 x_i（$i=1,\cdots,8$），建立如下的线性规划模型

$$\min \sum_{i\in V} x_i ,$$

$$\text{s.t.}\quad x_j \ge x_i + t_{ij},\ (i, j)\in A,\ i, j \in V, \qquad (4.1)$$

$$x_i \ge 0,\quad i\in V,$$

其中 V 是所有的事件集合，A 是所有作业的集合。

然后用下面的递推公式求其它指标。

$$z_n = x_n, (\text{这里} n=8)$$

$$z_i = \min_j \{z_j - t_{ij}\},\quad i = n-1,\cdots,1, (i, j)\in A \qquad (4.2)$$

$$es_{ij} = x_i,\ (i, j)\in A \qquad (4.3)$$

$$lf_{ij} = z_j,\ (i, j)\in A \qquad (4.4)$$

$$ls_{ij}=lf_{ij}-t_{ij}\,,\quad (i,j)\in A \tag{4.5}$$

$$ef_{ij}=x_i+t_{ij}\,,\quad (i,j)\in A \tag{4.6}$$

使用公式（4.3）和（4.5）可以得到所有作业的最早开工时间和最迟开工时间，如表 4.19 所示，方括号中第 1 个数字是最早开工时间，第 2 个数字是最迟开工时间。

表 4.19　作业数据

A	B	C	D	E	F	G	H
[0,0]	[0,11]	[6,6]	[9,9]	[11,11]	[5,16]	[14,14]	[18,18]

从表 4.19 可以看出，当最早开工时间与最迟开工时间相同时，对应的作业在关键路线上。关键路线为 1→2→4→5→6→7→8，关键路径的长度是 20 周。

计算的Lingo程序如下

```
model:
sets:
events/1..8/:x,z; !x为事件的最早时间，z为事件的最迟时间;
operate(events,events)/1 2,1 3,2 4,3 7,4 5,5 6,6 7,7 8/:t,s,ls,es,ef,lf;
!s为松弛变量，ls为作业的最迟开工时间，es为最早开工时间，ef为最早完工时间，lf为最迟完工时间;
endsets
data:
t=6 5 3 2 2 3 4 2;
@text(txt1.txt)=es,ls; !把计算结果输出到外部纯文本文件;
enddata
min=@sum(events:x);
@for(operate(i,j):x(j)>x(i)+t(i,j));
n=@size(events);
z(n)=x(n);
@for(events(i)|i#lt#n:z(i)=@min(operate(i,j):z(j)-t(i,j)));
@for(operate(i,j):es(i,j)=x(i));
@for(operate(i,j):lf(i,j)=z(j));
@for(operate(i,j):ls(i,j)=lf(i,j)-t(i,j));
@for(operate(i,j):ef(i,j)=x(i)+t(i,j));
end
```

（3）设 x_i 是事件 i 的开始时间，t_{ij} 是作业 (i,j) 的计划时间，m_{ij} 是完成作业 (i,j) 的最短时间， y_{ij} 是作业 (i,j) 可能减少的时间， c_{ij} 是作业 (i,j) 缩短一天增加的费用，因此有

$$x_j-x_i\ge t_{ij}-y_{ij}\quad 且\quad 0\le y_{ij}\le t_{ij}-m_{ij}\,.$$

17 周是要求完成的天数，1 为最初事件，8 为最终事件，所以有 $x_8-x_1\le 17$ 。而问题的总目标是使额外增加的费用最小，即目标函数为 $\min\sum\limits_{(i,j)\in A}c_{ij}y_{ij}$ 。由此得到相应的数学规划问题

$$\min\sum_{(i,j)\in A}c_{ij}y_{ij}\,,$$

s.t. $x_j - x_i + y_{ij} \ge t_{ij}$， $(i,j)\in A$， $i,j\in V$，

$x_n - x_1 \le 17$，

$0 \le y_{ij} \le t_{ij} - m_{ij}$， $(i,j)\in A$， $i,j\in V$.

求得作业C可以缩短一周，作业G可以缩短 2 周，额外支付的费用为 700 元。

计算的 Lingo 程序如下

```
model:
sets:
events/1..8/:x;
operate(events,events)/1 2,1 3,2 4,3 7,4 5,5 6,6 7,7 8/:t,m,c,y;
endsets
data:
t=6 5 3 2 2 3 4 2;
m=4 3 1 1 1 1 2 2;
c=800 600 300 300 600 400 200 200;
enddata
min=@sum(operate:c*y);
@for(operate(i,j):x(j)-x(i)+y(i,j)>t(i,j));
n=@size(events);
x(n)-x(1)<17;
@for(operate:@bnd(0,y,t-m));
end
```

（4）设t_{ij}， a_{ij}， e_{ij}， b_{ij}分别是完成作业(i,j)的实际时间（是一随机变量），最乐观时间，最可能时间，最悲观时间，通常用下面的方法计算相应的数学期望和方差，

$$E(t_{ij}) = \frac{a_{ij} + 4e_{ij} + b_{ij}}{6}, \tag{4.7}$$

$$\mathrm{var}(t_{ij}) = \frac{(b_{ij} - a_{ij})^2}{36}. \tag{4.8}$$

设T为实际工期，即

$$T = \sum_{(i,j)\in\text{关键路线}} t_{ij}, \tag{4.9}$$

由中心极限定理，可以假设T服从正态分布，并且期望值和方差满足

$$\bar{T} = E(T) = \sum_{(i,j)\in\text{关键路线}} E(t_{ij}), \tag{4.10}$$

$$S^2 = \mathrm{var}(T) = \sum_{(i,j)\in\text{关键路线}} \mathrm{var}(t_{ij}). \tag{4.11}$$

设规定的工期为d，则在规定的工期内完成整个项目的概率为

$$P\{T \le d\} = \Phi\left(\frac{d - \bar{T}}{S}\right). \tag{4.12}$$

对于这个问题采用最长路的方法。先按式（4.7）计算出各作业的期望值，再建立如下求关键路径的数学规划模型

$$\max \sum_{(i,j)\in A} E(t_{ij}) x_{ij},$$

$$\text{s.t.}\quad \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = \begin{cases} 1, & i=1, \\ -1, & i=n, \\ 0, & i\neq 1,n, \end{cases}$$

$$x_{ij}=0 \text{ 或 } 1,\ (i,j)\in A.$$

求出关键路径后，再由公式（4.8）计算出关键路线上各作业方差的估计值，最后利用式（4.12）即可计算出完成作业的概率与完成整个项目的时间。

计算得到关键路线的时间期望为 20.17 天，标准差为 1.77，产品在 21 周上市的概率为 68.1%，以 95%的概率完成新产品上市所需的周数为 24.1 周。

计算的 Lingo 程序如下

```
model:
sets:
events/1..8/:zd; !整数规划约束条件的常数项;
operate(events,events)/1 2,1 3,2 4,3 7,4 5,5 6,6 7,7 8/:a,e,b,et,dt,x;
endsets
data:
a=2  3  2  3  1  1  2  1;
e=6  5  3  4  2  3  4  2;
b=10 6 4 5 3 5 6 4;
zd=1 0 0 0 0 0 0 -1;
limit=21;
enddata
@for(operate:et=(a+4*e+b)/6;dt=(b-a)^2/36);
max=tbar;
tbar=@sum(operate:et*x);
@for(events(i):@sum(operate(i,j):x(i,j))-@sum(operate(j,i):x(j,i))=zd(i));
s^2=@sum(operate:dt*x);
p=@psn((limit-tbar)/s);
@psn((days-tbar)/s)=0.95;
end
```

4.8　解　记 v_i（$i=1,\cdots,5$）表示第 i 年年初的时刻，v_6 表示第 5 年末的时刻，构造赋权图 $G=(V,A,W)$，其中 $V=\{v_1,\cdots,v_6\}$， A 为弧的集合，邻接矩阵 $W=(w_{ij})_{6\times 6}$，这里 w_{ij} 为 v_i 到 v_j 的费用，包括购置费用和维修费用两部分。可以得到

$$W=\begin{bmatrix} 0 & 16 & 22 & 30 & 41 & 61 \\ \infty & 0 & 16 & 22 & 30 & 41 \\ \infty & \infty & 0 & 17 & 23 & 31 \\ \infty & \infty & \infty & 0 & 17 & 23 \\ \infty & \infty & \infty & \infty & 0 & 18 \\ \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

总费用最少的设备更新计划问题，归结为求图 G 中从 v_1 到 v_6 的费用最短路，可以使用 Dijkstra 标号算法求解。

求解的 Matlab 程序如下

```
clc, clear
a=zeros(6); %邻接矩阵初始化
a(1,[2:6])=[16 22 30 41 61]; %输入邻接矩阵，注意这里实际上为有向图
a(2,[3:6])=[16 22 30 41];
a(3,[4:6])=[17 23 31];
a(4,[5,6])=[17 23]; a(5,6)=18;
b=sparse(a); %有向图直接变成稀疏矩阵就可以了
[dist,path]=graphshortestpath(b,1,6,'Directed',1) %调用工具箱求最短路
```

求得的最优更新策略为第 3 年初更新设备，总费用为 53。

4.9 解 设 a_{ij}, b_{ij} 分别表示工作 (i, j) 的正常工时和特急工时，c_{ij}, d_{ij} 分别表示工作 (i, j) 的正常费用和特急费用，x_i 表示事件 i 开始的时间，y_{ij} 表示工作 (i, j) 可能减少的天数。

总的费用包括三部分，间接费用 $15(x_8 - x_1)$，正常费用 $\sum_{(i,j)\in A} c_{ij}(a_{ij} - y_{ij})$，特急费用 $\sum_{(i,j)\in A} d_{ij} y_{ij}$，因而建立如下的数学规划模型

$$\min \quad 15(x_8 - x_1) + \sum_{(i,j)\in A} (d_{ij} - 2c_{ij}) y_{ij} + \sum_{(i,j)\in A} c_{ij} a_{ij}$$

$$\text{s.t.} \quad x_j - x_i \geq a_{ij} - y_{ij}, \quad (i, j) \in A, \quad i, j \in V,$$

$$0 \leq y_{ij} \leq a_{ij} - b_{ij}, \quad (i, j) \in A, \quad i, j \in V,$$

$$x_1 = 0 .$$

求得总的工期为 27 天，总费用为 7805 元。

计算的 Lingo 程序如下

```
model:
sets:
events/1..8/:x;
operate(events,events)/1 2,2 3,2 4,3 4,3 5,4 6,4 7,5 8,6 8,7 8/:a,b,c,d,y;
endsets
data:
a=6 9 3 0 7 8 2 1 4 5;
b=4 5 2 0 5 3 1 1 3 2;
c=100 200 80 0 150 250 120 100 180 130;
d=120 280 110 0 180 375 170 100 200 220;
enddata
min=15*(x(8)-x(1))+@sum(operate:(d-c)*y+c*a);
@for(operate(i,j):x(j)-x(i)>a(i,j)-y(i,j));
@for(operate:@bnd(0,y,a-b));
x(1)=0;
end
```