第七章代码部分

钱昌发

2019 年 6 月 9 日

1 代码部分

由于本节代码较多，现将所有代码以函数方式给出，待到解题的时候直接调用函数就可以了：

1.1 图论代码

Dijkstra 算法的 Matlab 函数：

```
function [d Q] = shorta(T)
pp(1:length(T)) = 0; pp(1) = 1; Q = 1;
M = max(T(:)); d(1:length(T)) = M; d(1) = 0; K = 1;
while sum(pp)<length(T)
  tt = find(pp==0); % 找出未标记的点
  d(tt) = min(d(tt), d(K)+T(K,tt));
  ttt = find(d(tt)==min(d(tt)));
  K = tt(ttt(1)); pp(K) = 1; Q = [Q, K];
end
```

Floyd 算法的 Matlab 函数：

```
function [P, u] = f_path(W)
% W 表示权值矩阵; P 表示最短路; % u 表示最短路的权和
n = length(W); U = W; k = 1; % Step1 初始化
% Step2
while k<=n
  for i=1:n
    for j=1:n
      if U(i, j) > U(i, k) + U(k, j)
        U(i, j) = U(i, k) + U(k, j);
      end;
```

```
    end;
  end
k = k+1;
end
u = U(1, n);
% 输出最短路的顶点
P1 = zeros(1,n); k = 1; P1(k) = n; V = ones(1,n)*inf; kk = n;
while kk~=1
  for i=1:n
    V(1, i) = U(1, kk) - W(i, kk);
    if V(1, i)==U(1, i)
      P1(k+1) = i; kk = i; k = k+1;
    end;
  end;
end
k = 1; wrow = find(P1~=0);
for j=length(wrow) : (-1) : 1
  P(k) = P1(wrow(j)); k = k+1;
end
```

0-1 规划模型算法：

```
function y=op01(W)
%0- -1 规划模型的MATLAB 程序
n = length(W);
A = zeros(n, n*n);
intcon=1:n*n;
for i = 1:n
  e1 = zeros(1, n);
  e1(i) = 1;
  e2 = -1*ones(1, n);
  e2(i) = 0;
  A(i, :) = repmat(e1, 1, n);
  A(i, (i-1)*n+1:i*n) = e2;
end
b = zeros(n, 1);
b(1) = 1;
b(end) = -1;
lb=zeros(n*n,1);
ub=ones(n*n,1);
x = intlinprog(W,intcon,[],[],A,b,lb,ub);
y = reshape(x, n, n);
```

1.2 网络流模型代码

1.2.1 最大流模型代码

Ford—Fulkerson 算法代码：

```
function f=ford(u,f)
%Ford—Fulkerson 算法的Matlab
n = length(u); list = [ ]; maxf = zeros(1:n); maxf(n) = 1;
M=1000;
while maxf(n)>0
  maxf = zeros(1, n); pred=zeros(1, n);
  list = 1; record = list; maxf(1) = M;
  while (~isempty(list)) & (maxf(n)==0)
    flag = list(1); list(1) = []; index1 = (find(u(flag, :)~=0));
    label1 = index1(find(u(flag, index1) - f(flag, index1)~=0));
    label1 = setdiff(label1, record); list = union(list, label1);
    pred(label1(find(pred(label1)==0))) = flag;
    maxf(label1) = min(maxf(flag), u(flag, label1) - f(flag, label1));
    record = union(record, label1); label2 = find(f(:, flag)~=0);
    label2 = label2'; label2 = setdiff(label2,record);
    list = union(list, label2);
    pred(label2(find(pred(label2)==0))) = -flag;
    maxf(label2) = min(maxf(flag), f(label2, flag));
    record = union(record, label2);
  end
  if maxf(n)>0
    v2 = n; v1 = pred(v2);
    while v2~=1
      if v1>0
        f(v1,v2) = f(v1, v2)+maxf(n);
      else
        v1 = abs(v1); f(v2, v1) = f(v2, v1)-maxf(n);
      end
      v2 = v1; v1 = pred(v2);
    end;
  end;
end
f; % 最后的f为最大流量矩阵
```

规划模型的代码：

```
function x=op02(u)
n =length(u);
```

```
e = [1, zeros(1, n-1)]; c = repmat(-e, 1, n);
A = repmat(e, 1, n); A(end-n+1:end) = A(end-n+1:end) - 1;
for i = 2:n-1
  e1 = zeros(1, n); e1(i) = 1; e2 = -1*ones(1, n); e2(i) = 0;
  A(i,:) = repmat(e1, 1, n); A(i,(i-1)*n+1:i*n) = e2;
end
b = zeros(n-1,1);
intcon=1:36;
[x, f ] = intlinprog(c,intcon, [ ], [ ], A, b, zeros(n*n, 1), u(:));
x = reshape(x, n, n); % 最后的f
```

1.2.2 最小费用最大流模型

最小费用最大流模型 Ford 算法代码：

```
function [f,wf,zwf]=ford02(C,b)
%最小费用最大流问题的Matlab 代码
%C是弧容量
%b是费用
n = length(C);
wf = 0; wf0 = Inf; % wf 表示最大流量, wf0 表示预定的流量值
f = zeros(n,n); % 取初始可行流f 为零流
while 1
  for i=1:n
    for j=1:n
      if (j~=i)
        a(i,j) = inf;
      end;
    end;
  end % 构造有向赋权图
  for i=1:n
    for j=1:n
      if (C(i,j)>0 & f(i,j)==0)
        a(i,j) = b(i,j);
      elseif (C(i,j)>0 & f(i,j)==C(i,j))
        a(j,i) = -b(i,j);
      elseif (C(i,j)>0)
        a(i,j) = b(i,j); a(j,i) = -b(i,j);
      end
    end
  end
  for i=2:n
```

```
  p(i) = inf; s(i) = i;
end % 用Ford 算法求最短路, 赋初值
for (k=1:n)
  pd = 1; % 求有向赋权图中vs 到vt 的最短路
  for (i=2:n)
    for (j=1:n)
      if (p(i)>p(j)+a(j,i))
        p(i) = p(j)+a(j,i); s(i) = j; pd = 0;
      end;
    end;
  end
  if (pd)
    break;
  end;
end % 求最短路的Ford 算法结束
if (p(n)==inf)
  break;
end % 不存在vs 到vt 的最短路, 算法终止. 注意在求最小费
% 用最大流时构造有向赋权图中不含负权回路, 故不出现k=n
dvt = inf; t=n; % 进入调整过程, dvt 表示调整量
while (1) % 计算调整量
  if (a(s(t), t)>0)
    dvtt = C(s(t), t)-f(s(t), t); % 前向弧调整量
  elseif (a(s(t), t)<0)
    dvtt = f(t, s(t)); % 后向弧调整量
  end
  if (dvt>dvtt)
    dvt = dvtt;
  end
  if (s(t)==1)
    break;
  end % 当t 的标号为vs 时, 终止计算调整量
  t = s(t);
  end % 继续调整前一段弧上的流f
  pd = 0;
  if (wf+dvt>=wf0)
    dvt = wf0-wf; pd = 1;
  end % 如果最大流量大于或等于预定的流量值
  t = n;
while (1) % 调整过程
  if (a(s(t), t)>0)
    f(s(t), t) = f(s(t), t)+dvt; % 前向弧调整
```

```
    elseif (a(s(t), t)<0)
      f(t,s(t)) = f(t,s(t))-dvt; % 后向弧调整
    end
    if (s(t)==1)
      break;
    end % 当t 的标号为vs 时, 终止调整过程
    t = s(t);
  end
  if (pd)
    break;
  end % 如果最大流量达到预定的流量值
  wf = 0;
  for (j=1:n)
    wf = wf+f(1, j);
  end;
end % 计算最大流量
zwf = 0;
for (i=1:n)
  for (j=1:n)
    zwf = zwf+b(i, j)*f(i, j);
  end
end % 计算最小费用
```

最小费用最大流规划算法代码：

```
function [f,wf]=op03(C,w)
n = length(C);
e = [1, zeros(1, n-1)]; c = repmat(-e, 1, n);
A = repmat(e, 1, n); A(end-n+1:end) = A(end-n+1:end) - 1;
for i = 2:n-1
  e1 = zeros(1, n); e1(i) = 1; e2 = -1*ones(1, n); e2(i) = 0;
A(i,:) = repmat(e1, 1, n); A(i,(i-1)*n+1:i*n) = e2;
end
b = zeros(n-1,1);
intcon=1:n*n;
[x, fv ] = intlinprog(c, intcon,[ ], [ ], A, b, zeros(n*n, 1), C(:));
f = reshape(x, n, n);
A = repmat(e, 1, n);
for i = 2:n
  e1 = zeros(1, n); e1(i) = 1; e2 = -1*ones(1, n); e2(i) = 0;
  A(i,:) = repmat(e1, 1, n); A(i,(i-1)*n+1:i*n) = e2;
end
b = [-fv; zeros(n-2,1); fv ];
```

```
[x, gv ] = linprog(w, [ ], [ ], A, b, zeros(n*n, 1), C(:));
wf = reshape(x, n, n); % 最小费用最大流量矩阵
```

1.3 最优连线模型与最优环游模型代码

1.3.1 最小生成树代码

避圈法代码：

```
function A = avoidcircle(W)
[m, n] = size(W);
e = 0;
for i = 1 : n
  for j = i : n
    if W(i, j) ~= 0
      e = e + 1;
      E(e, :) = [i, j, W(i, j)];
    end
  end
end
% 按权值大小排列边的顺序
for i = 1 : e - 1
  for j = i + 1 : e
    if E(i, 3) > E(j, 3)
      temp = E(j, :);
      E(j, :) = E(i, :);
      E(i, :) = temp;
    end
  end
end
A = zeros(1, 3); S = 1 : n;
for i = 1 : e
  if S(E(i, 1)) ~= S(E(i, 2))
    A = cat(1, A, E(i,:));
    indicator = S(E(i, 1));
    for j = 1 : n
      if S(j) == indicator
        S(j) = S(E(i, 2));
      end
    end
  end
end
```

```
A(1, :) = [];
```

破圈法代码：
暂时空着：

1.3.2 最优环游模型：

改良圈算法代码：

```
function [circle,sum]=circle1(a)
a = a+a';
c1 = [5 1:4 6];
L = length(c1);
flag = 1;
while flag>0
  flag = 0;
  for m=1:L-3
    for n=m+2:L-1
      if a(c1(m),c1(n))+a(c1(m+1),c1(n+1))< a(c1(m),c1(m+1))+a(c1(n),c1(n+1))
        flag = 1;
        c1(m+1:n) = c1(n:-1:m+1);
      end;
    end;
  end;
end
sum1 = 0;
for i=1:L-1
  sum1 = sum1+a(c1(i),c1(i+1));
end
circle = c1;
sum = sum1;
c1 = [5 6 1:4]; % 改变初始圈，最后一个顶点不动
sum1 = 0; flag = 1;
while flag>0
  flag=0;
  for m=1:L-3
    for n=m+2:L-1
      if a(c1(m),c1(n))+a(c1(m+1),c1(n+1)) < ...
        a(c1(m),c1(m+1))+a(c1(n),c1(n+1))
        flag=1; c1(m+1:n)=c1(n:-1:m+1);
      end;
    end;
```

```
    end;
  end
  sum1 = 0;
  for i=1:L-1
    sum1 = sum1+a(c1(i),c1(i+1));
  end
  if sum1<sum
    sum = sum1;
    circle = c1;
  end
```

规划算法代码:

```
%此为错误代码, 待修正
  function x=op04(a)
  n = length(a); a = a+a';
  A = kron(eye(n), ones(1, n));
  A(n+1:2*n, :) = repmat(eye(n), 1, n);
  b = ones(2*n, 1);
  intcon=1:36;
  [x, f ] = intlinprog(a(:),intcon, [ ], [ ], A, b,zeros(36,1),ones(36,1));
  x = reshape(x, n, n);
```

2 题目解答:

2.1 第一题答案:

建立图论矩阵:

$$\mathbf{A}=\begin{bmatrix} 0 & 1 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 3 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 5 & 7 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

然后直接作为矩阵带入函数即可，但是需要注意的是带入时将零换成极大值即可，求解代码:

```
%(1)Dijkstra算法:
  m=10000;
```

```
  W=[0,1,4,m,m,m,m;
     m,0,5,3,5,m,m;
     m,m,0,m,m,2,m;
     m,m,m,0,5,7,3;
     m,m,m,m,0,m,7;
     m,m,m,m,m,0,2;
     m,m,m,m,m,m,0;];
  [d,Q]=shorta(W)
  %answer
  d =
     0 1 4 4 6 6 7
  Q =
     1 2 3 4 5 6 7
%(2)Floyd算法:
  [d,Q]=f_path(W)
  %answer
  d =
    1     2     4     7
  Q =
    7
%(3)规划算法:
  d=op01(W)
  %answer
  LP:Optimal objective value is 7.000000.
  d =
  0 1 0 0 0 0 0
  0 0 0 1 0 0 0
  0 0 0 0 0 0 0
  0 0 0 0 0 0 1
  0 0 0 0 0 0 0
  0 0 0 0 0 0 0
  0 0 0 0 0 0 0
```

2.2 第二题答案:

解法同上，求解代码:

```
%(1)Dijkstra算法:
m=10000;
W=[0,9,8,m,m,m,m;
   m,0,5,2,1,m,m;
   m,m,0,8,m,7,m;
```

```
   m,m,m,0,2,3,m;
   m,m,m,m,0,m,3;
   m,m,m,m,m,0,4;
   m,m,m,m,m,m,0;];
   [d,Q]=shorta(W)
   %answer
   d =
     0 9 8 11 10 14 13
   Q =
     1 3 2 5 4 7 6
%(2)Floyd算法:
  [d,Q]=f_path(W)
  %answer
  d =
    1 2 5 7
  Q =
    7
%(3)规划算法:
  d=op01(W)
  %answer
  LP:Optimal objective value is 13.000000.
  d =
     0 1 0 0 0 0 0
     0 0 0 1 0 0 0
     0 0 0 0 0 0 0
     0 0 0 0 0 0 0
     0 0 0 0 0 0 1
     0 0 0 0 0 0 0
     0 0 0 0 0 0 0
```

2.3 第三题答案:

求解代码:

```
  f=zeros(6,6);
  u=[0,16,20,0,0,0,;
  0,0,0,10,0,10;
  0,0,0,6,6,0;
  0,0,0,0,0,10;
  0,0,0,0,0,16;
  0,0,0,0,0,0;];
  f=ford(u,f)
```

```
%answer:
f =
  0 16 10 0 0 0
  0 0 0 6 0 10
  0 0 0 4 6 0
  0 0 0 0 0 10
  0 0 0 0 0 6
  0 0 0 0 0 0
```

2.4 第四题答案：

求解代码：

```
f=zeros(7,7);
u=[0,7,8,6,0,0,0;
   0,0,0,0,5,0,0;
   0,3,0,2,5,3,0;
   0,0,0,0,0,10;
   0,0,0,0,3,0,9;
   0,0,0,0,0,0;];
f=ford(u,f)
%answer:
f =
  0 5 8 5 0 0 0
  0 0 0 0 5 0 0
  0 0 0 0 5 3 0
  0 0 0 0 0 5 0
  0 0 0 0 0 0 10
  0 0 0 0 0 0 8
  0 0 0 0 0 0 0
```

2.5 第五题答案：

求解代码：

```
f=-[2,3,4,1,7;
3,4,2,5,6;
2,5,3,4,1;
5,2,3,2,5;
3,7,6,2,4];
intcon=1:25;
A=[];
```

```
b=[];
Aeq=[ones(1,5),zeros(1,20);
zeros(1,5),ones(1,5),zeros(1,15);
zeros(1,10),ones(1,5),zeros(1,10);
zeros(1,15),ones(1,5),zeros(1,5);
zeros(1,20),ones(1,5);
full(sparse(ones(1,5),[1,6,11,16,21],ones(1,5))),zeros(1,4);
full(sparse(ones(1,5),[2,7,12,17,22],ones(1,5))),zeros(1,3);
full(sparse(ones(1,5),[3,8,13,18,23],ones(1,5))),zeros(1,2);
full(sparse(ones(1,5),[4,9,14,19,24],ones(1,5))),zeros(1,1);
full(sparse(ones(1,5),[5,10,15,20,25],ones(1,5)))];
beq=ones(10,1);
lb=zeros(25,1);
ub=ones(25,1);
[a,z]=intlinprog(f,intcon,A,b,Aeq,beq,lb,ub);
a=reshape(a,[5,5])
%answer:
a=
  0 0 0 0 1
  0 0 0 1 0
  0 1 0 0 0
  1 0 0 0 0
  0 0 1 0 0
```

2.6 第六题答案:

求解代码:

```
C = [0,6,2,1,0;0,0,0,0,0;0,2,0,10,3;0,4,0,0,0;0,0,0,0,0]; % 弧容量
b = [0,5,9,4,0;0,0,0,0,0;0,3,0,4,2;0,3,0,0,0;0,0,0,0,0];
[f,wf,zwf]=ford02(C,b)
%answer:
f =
  0     0     2     0     0
  0     0     0     0     0
  0     0     0     0     2
  0     0     0     0     0
  0     0     0     0     0
wf =
  2
zwf =
  22
```

2.7 第七题答案:

求解代码:

```
C = [0,2,8,0,0,0;
     0,0,5,2,0,0;
     0,0,0,0,3,0;
     0,0,1,0,0,6;
     0,0,0,4,0,7;
    0,0,0,0,0,0;]; % 弧容量
b = [0,8,7,0,0,0;
     0,0,5,9,0,0;
     0,0,0,0,9,0;
     0,0,2,0,0,5;
     0,0,0,6,0,10;
     0,0,0,0,0,0];
[f,wf,zwf]=ford02(C,b)
%answer:
f =
  0     2     3     0     0     0
  0     0     0     2     0     0
  0     0     0     0     3     0
  0     0     0     0     0     2
  0     0     0     0     0     3
  0     0     0     0     0     0
wf =
5
zwf =
122
```

2.8 第八题答案:

求解代码:

```
a(1, 1:6) = [0,3, 7, 4, 0, 0];
a(2, 1:6) = [3, 0, 2,0,9,0];
a(3, 1:6) = [7,2,0,1,6,3];
a(4, 1:6) = [4,0,1,0,0,4];
a(5, 1:6) = [0,9,6,0,0,3]; a(6, :)=0;
aviodcircle(a)
%answer:
ans =
     3     4     1
```

```
     2     3     2
     1     2     3
     3     6     3
     5     6     3
```

2.9 第九题答案:

求解代码:

```
a(1,1:9)=[0,2,1,3,0,0,0,0,0];
a(2,1:9)=[2,0,4,0,5,6,0,0,0];
a(3,1:9)=[1,4,0,3,5,0,0,0,0];
a(4,1:9)=[3,0,5,0,6,0,0,8,0];
a(5,1:9)=[0,5,3,6,0,4,0,0,0];
a(6,1:9)=[0,2,0,0,4,0,5,0,3];
a(7,1:9)=[0,0,0,0,3,5,0,4,1];
a(8,1:9)=[0,0,0,8,7,0,4,0,2];
a(9,1:9)=[0,0,0,0,0,0,0,0,0];
avoidcircle(W)
%answer:
ans =

     2     5     1
     2     4     2
     4     6     3
     5     7     3
     2     3     5
     1     3     8
```

2.10 第十题答案:

求解代码:

```
a(1,2)=10;a(1,3)=20;a(1,4)=30;a(1,5)=40;a(1,6)=50;
a(2,3)=18;a(2,4)=30;a(2,5)=25;a(2,6)=21;
a(3,4)=5;a(3,5)=10;a(3,6)=15;
a(4,5)=8;a(4,6)=16;
a(5,6)=18;
a(6,:)=0;
[circle,sum]=circle1(a)
%answer:
circle =
```

```
5     4     3     1     2     6
sum =
64
```