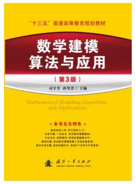
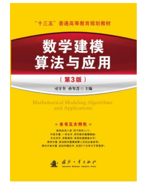


数学建模算法与应用

第4章

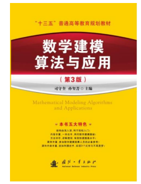
4.5着色问题

4.8旅行商（TSP）问题



数学建模算法与应用

目录 CONTENTS



数学建模算法与应用

01 着色问题

已知图$G=(V,E)$，对图G的所有顶点进行着色时要求相邻的两顶点的颜色不一样，问至少需要几种颜色？这就是所谓的顶点着色问题。

若对图G的所有边进行着色时，要求相邻的两条边的颜色不一样，问至少需要几种颜色？这就是所谓的边着色问题。

这些问题的提出是有实际背景的。值得注意的是着色模型中的图是无向图。对于边着色问题可以转化为顶点着色问题。

例 4.14 物资存储问题。一家公司制造n种化学制品$A_1,A_2,\cdots,A_n$，其中有些化学制品若放在一起可能产生危险，如引发爆炸或产生毒气等，称这样的化学制品是不相容的。为安全起见，在存储这些化学制品时，不相容的不能放在同一储存室内。问至少需要多少个存储室才能存放这些化学制品?

构造图G，用顶点$v_1, v_2, \cdots, v_n$分别表示n种化学制品，顶点v_i与v_j相邻，当且仅当化学制品A_i与A_j不相容。

于是存储问题就化为对图G的顶点着色问题，对图G的顶点最少着色数目便是最少需要的储存室数。

例 4.15 无线交换设备的波长分配。

有 5 台设备，要给每一台设备分配一个波长。如果两台设备靠得太近，则不能给它们分配相同的波长以防干扰。已知v_1和v_2,v_4,v_5靠得近，v_2和v_3,v_4靠得近，v_3和v_4,v_5靠得近。问至少需要几个发射波长。

以设备为顶点构造图 $G=(V,E)$，其中 $V=\{v_1,v_2,\cdots,v_5\}$，$v_1,v_2,\cdots,v_5$ 分别代表5台设备。E 为边集，如果两台设备靠得太近，则用一条边连接它们。于是图 G 的着色给出一个波长分配方案：给着同一种颜色的设备同一个波长。画出着色图如图4.13所示，可知需要3个发射波长。

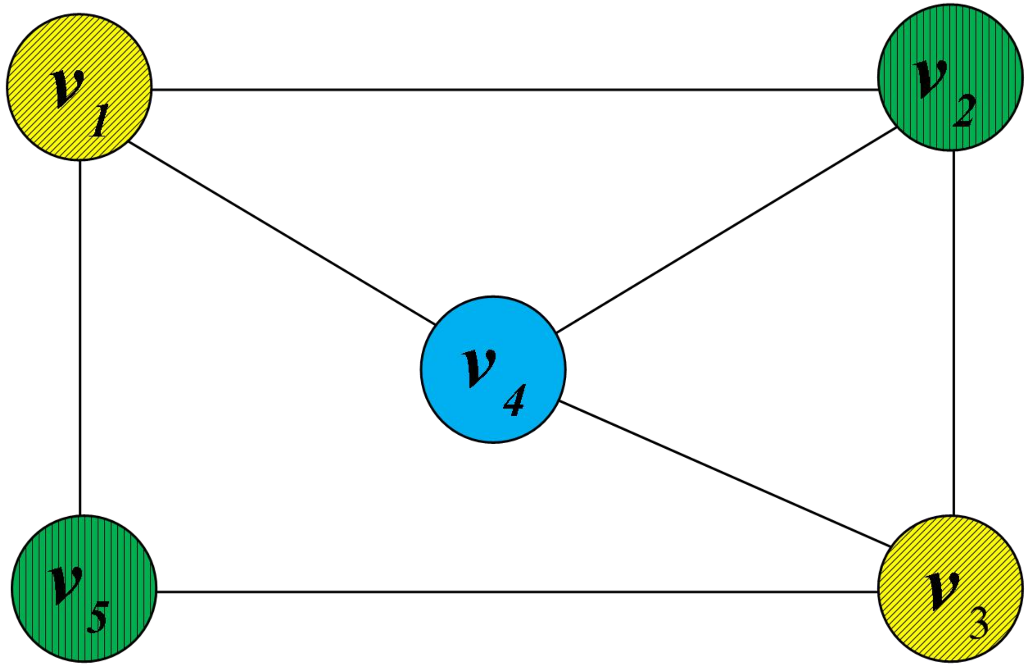


图 4.13　5 台设备的关系图

对图$G=(V,E)$的顶点进行着色所需最少的颜色数目用$\chi(G)$表示，称为图G的色数。

定理 4.4 若图$G=(V,E)$，$\Delta=\max\{d(v)\mid v\in V\}$为图$G$顶点的最大度数，则$\chi(G)\le\Delta+1$。

例 4.16 会议安排

学校的学生会下设 6 个部门，部门的成员如下：部门 1={张，李，王}，部门 2={李，赵，刘}，部门 3={张，刘，王}，部门 4={赵，刘，孙}，部门 5={张，王，孙}，部门 6={李，刘，王}，每个月每个部门都要开一次会，为了确保每个人都能参加他所在部门的会议，这 6 个会议至少需要安排在几个不同的时段？

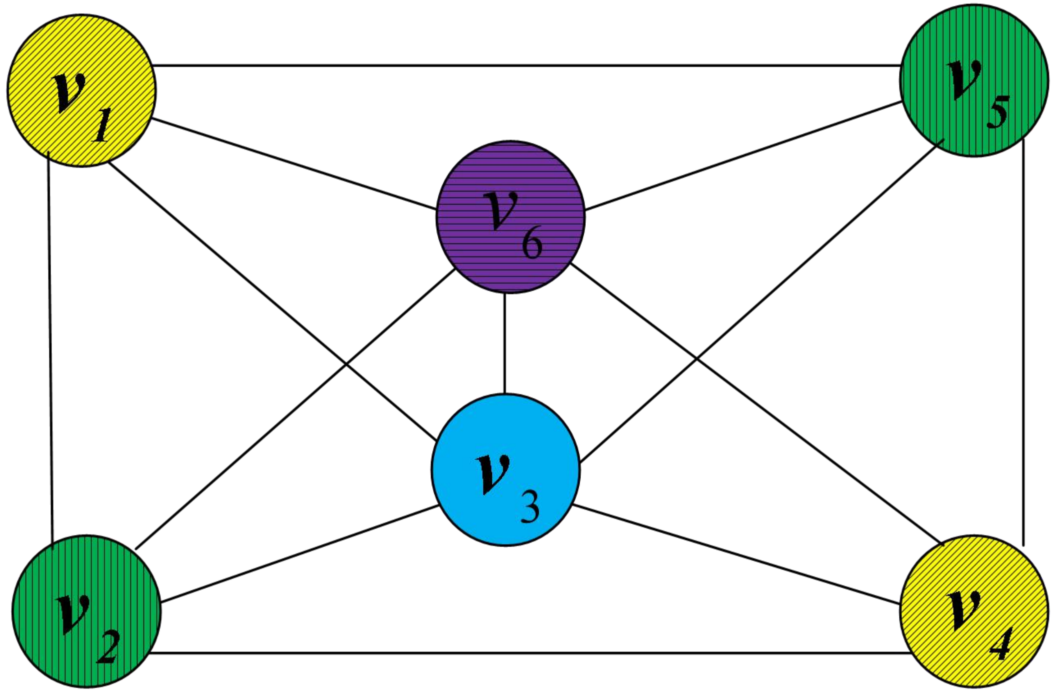


图 4.14 部门之间关系图

构造图$G=(V,E)$，其中$V=\{v_1,v_2,\cdots,v_6\}$，这里$v_1,v_2,\cdots,v_6$分别表示部门1，部门2，…，部门6；E为边集，两个顶点之间有一条边当且仅当它们代表的委员会成员中有共同的人，如图4.14所示，该图可以用4种颜色着色，可以看出至少要用4种颜色，v_1,v_2,v_3构成一个三角形，必须用3种颜色，v_6和这3个顶点都相邻，必须再用一种颜色。

着同一种颜色的顶点代表的部门会议可以安排在同一时间段，而不同颜色代表的部门会议必须安排在不同的时间，故这6个会议至少要安排在4个不同的时间，其中，部门1和部门4，部门2和部门5的会议可以安排在同一时间段。

定理 4.4 给出了色数的上界，着色算法目前还没有找到最优算法。下面给出例 4.16 计算色数的整数线性规划模型。

例 4.16 中顶点个数$n=6$，顶点的最大度$\Delta=5$。

引入$0-1$变量

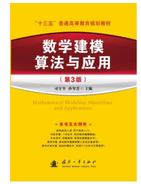
$$x_{ik}=\begin{cases}1, & \text{当}v_i\text{着第}k\text{种颜色时}, \\ 0, & \text{否则},\end{cases} \quad i=1,2,\cdots,n;\ k=1,2,\cdots$$

设颜色总数为y，建立如下整数线性规划模型：

$$\min y,$$

$$\text{s.t.}\begin{cases}\sum_{k=1}^{\Delta+1} x_{ik}=1, & i=1,2,\cdots,n,\\ x_{ik}+x_{jk}\le 1, & (v_i,v_j)\in E, k=1,2,\cdots,\Delta+1,\\ y\ge\sum_{k=1}^{\Delta+1} kx_{ik}, & i=1,2,\cdots,n,\\ x_{ik}=0\text{或}1, & i=1,2,\cdots,n, k=1,2,\cdots,\Delta+1.\end{cases}$$

```
clc, clear
s={{'张','李','王'};{'李','赵','刘'};{'张','刘','王'};
   {'赵','刘','孙'};{'张','王','孙'};{'李','刘','王'}};
n = length(s); w = zeros(n);
for i = 1:n-1
   for j =i+1:n
      if ~isempty(intersect(s{i},s{j}))
         w(i,j)=1;
      end
   end
end
[ni,nj] = find(w);  %边的顶点编号
w = w + w'; %计算完整的邻接矩阵
deg = sum(w); K = max(deg) %顶点的最大度
prob = optimproblem;
x = optimvar('x',n,K+1, 'Type','integer','LowerBound',0,'UpperBound',1);
y = optimvar('y'); prob.Objective = y;
prob.Constraints.con1 = sum(x,2)==1;
prob.Constraints.con2 = x(ni,:)+x(nj,:)<=1;
prob.Constraints.con3 = x*[1:K+1]'<=y;
[sol, fval, flag, out] = solve(prob)
[i,k] = find(sol.x);
fprintf('顶点和颜色的对应关系如下：\n')
ik = [i'; k']
```



02 旅行商（TSP）问题

4.8.1 修改圈近似算法

一名推销员准备前往若干城市推销产品，然后回到他的出发地。如何为他设计一条最短的旅行路线（从驻地出发，经过每个城市恰好一次，最后返回驻地）？这个问题称为旅行商问题。用图论的术语说，就是在一个赋权完全图中，找出一个有最小权的Hamilton圈。称这种圈为最优圈。目前还没有求解旅行商问题的有效算法。所以希望有一个方法以获得相当好（但不一定最优）的解。

一个可行的办法是首先求一个 Hamilton 圈 C，然后适当修改 C 以得到具有较小权的另一个 Hamilton 圈。修改的方法叫做改良圈算法。设初始圈 $C = v_1 v_2 \cdots v_n v_1$。

(1) 对于$1\le i<i+1<j\le n$，构造新的 Hamilton 圈

$$C_{ij}=v_1v_2\cdots v_iv_jv_{j-1}v_{j-2}\cdots v_{i+1}v_{j+1}v_{j+2}\cdots v_nv_1,$$

它是由C中删去边v_iv_{i+1}和v_jv_{j+1}，添加边v_iv_j和$v_{i+1}v_{j+1}$而得到的。若

$$w(v_iv_j)+w(v_{i+1}v_{j+1})<w(v_iv_{i+1})+w(v_jv_{j+1}),$$

则以C_{ij}代替C，C_{ij}叫做C的改良圈。

(2) 转 (1)，直至无法改进，停止。

用改良圈算法得到的结果几乎可以肯定不是最优的。为了得到更高的精确度，可以选择不同的初始圈，重复进行几次，以求得较精确的结果。

圈的修改过程一次替换三条边比一次仅替换两条边更有效；然而，有点奇怪的是，进一步推广这一想法，就不对了。

例 4.21 从北京(Pe)乘飞机到东京(T)、纽约(N)、墨西哥城(M)、伦敦(L)、巴黎(Pa)五城市做旅游，每城市恰去一次再回北京，应如何安排旅游线，使旅程最短？用修改圈算法，求一个近似解。各城市之间的航线距离如表 4.6。

表 4.6　六城市间的距离

	L	M	N	Pa	Pe	T
L		56	35	21	51	60
M	56		21	57	78	70
N	35	21		36	68	68
Pa	21	57	36		51	61
Pe	51	78	68	51		13
T	60	70	68	61	13	

解　求得近似圈为 5→4→1→3→2→6→5；近似圈的长度为211。

实际上可以用数学规划模型求得精确的最短圈长度为211，这里的近似算法凑巧求出了准确解。

```
clc, clear, a=readmatrix ('data4_21.xlsx');
a(isnan(a))=0  %对角线元素替换为0
L=size(a,1); c=[5 1:4 6 5]; %选取初始圈
[circle,long]=modifycircle(a,L,c)  %调用下面修改圈的子函数
function [circle,long]=modifycircle(a,L,c)
for k=1:L
   flag=0;   %退出标志
   for i=1:L-2
      for j=i+2:L
         if a(c(i),c(j))+a(c(i+1),c(j+1))<...
               a(c(i),c(i+1))+a(c(j),c(j+1))
            c(i+1:j)=c(j:-1:i+1);
            flag=flag+1; %修改一次，标志加1
         end
      end
   end
   if flag==0   %一条边也没有修改,就返回
      long=0;   %圈长的初始值
      for i=1:L
         long=long+a(c(i),c(i+1)); %求改良圈的长度
      end
      circle=c;   %返回修改圈
      return
   end
end
end
```

4.8.2旅行商问题的数学规划模型

旅行商的0–1整数规划模型在第 2 章整数规划中已经给出了，这里就不赘述了。

例 4.22 已知 SV 地区各城镇之间距离见表 4.7，某公司计划在 SV 地区做广告宣传，推销员从城市 1 出发，经过各个城镇，再回到城市 1。为节约开支，公司希望推销员走过这 10 个城镇的总距离最少。

表 4.7 城镇之间的距离

	2	3	4	5	6	7	8	9	10
1	8	5	9	12	14	12	16	17	22
2		9	15	17	8	11	18	14	22
3			7	9	11	7	12	12	17
4				3	17	10	7	15	18
5					8	10	6	15	15
6						9	14	8	16
7							8	6	11
8								11	11
9									10

解　求得的最短路径为

$$1\to3\to7\to9\to10\to8\to4\to5\to6\to2\to1,$$

最短路径长度为 73。

```
clc, clear, a=readmatrix('data4_22.xlsx');
a(isnan(a))=0; %把NaN替换为0
b=zeros(10); %邻接矩阵初始化
b([1:end-1],[2:end])=a; %邻接矩阵上三角元素赋值
b=b+b'; %构造完整的邻接矩阵
n=10; b([1:n+1:end])=1000000; %对角线元素换为充分大
prob=optimproblem;
x=optimvar('x',n,n,'Type','integer','LowerBound',0,'UpperBound',1);
u=optimvar('u',n,'LowerBound',0)  %序号变量
prob.Objective=sum(sum(b.*x));
prob.Constraints.con1=[sum(x,2)==1; sum(x,1)'==1; u(1)==0];
con2 = [1<=u(2:end); u(2:end)<=14];
for i=1:n
    for j=2:n
        con2 = [con2; u(i)-u(j)+n*x(i,j)<=n-1];
    end
end
prob.Constraints.con2 = con2;
[sol, fval, flag]=solve(prob)
xx=sol.x; [i,j]=find(xx);
fprintf('xij=1对应的行列位置如下：\n')
ij=[i'; j']
```