**2.2**

**问题分析**

这是常见的0-1决策问题，需要我们在最小建校时能覆盖最大的区域

**符号说明**

:对第i(i=1,2,3,4,5,6)个区域的选取

**模型建立**

对小区A1我们有



对小区A2，我们有



对小区A3，我们有



对小区A4，我们有



对小区A5，我们有



对小区A6，我们有



对小区A7，我们有



对小区A8，我们有



于是，我们只要求解



**代码如下**：

clc,clear

prob=optimproblem;

x=optimvar('x',6,'Type','integer','LowerBound',0,'UpperBound',1);

prob.Objective = sum(x);

prob.Constraints.con1=x(1)+x(2)+x(3)>=1;

prob.Constraints.con2=x(2)+x(4)>=1;

prob.Constraints.con3=x(3)+x(5)>=1;

prob.Constraints.con4=x(4)+x(6)>=1;

prob.Constraints.con5=x(6)+x(5)>=1;

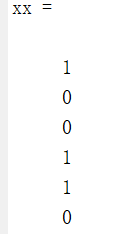
prob.Constraints.con6=x(1)>=1;

prob.Constraints.con7=x(4)+x(2)+x(6)>=1;

prob.Constraints.con8=x(2)+x(5)>=1;

[sol,fval,flag]=solve(prob),xx=sol.x

**结果如下**：



即此时只要对B1, B4, B5,建设即可

2.3

**问题分析**

这是常见的0-1决策问题，需要我们在限定设备时做出能提供最大收益的决策

**符号说明**

:对第i(i=1,2,3,4)个企业对第j(j=1,2,3,4)个工厂的选取

：:所对因的盈利

**模型建立**

每个企业至少有一个设备，我们有



每个设备都有一台，我们有



于是问题转化为



**代码如下**：

clc,clear

prob=optimproblem('ObjectiveSense','max');

b=[4,2,3,4;6,4,5,5;7,6,7,6;7,8,8,6;7,9,8,6;7,10,8,6]

x=optimvar('x',6,4,'Type','integer','LowerBound',0,'UpperBound',1);

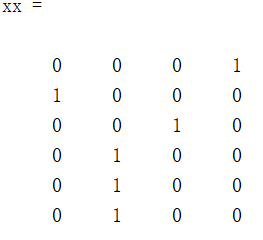
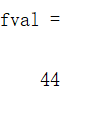
prob.Objective = sum(b.\*x,'all');

prob.Constraints.con1=sum(x,2)==1;

prob.Constraints.con2=sum(x,1)>=1;

[sol,fval,flag]=solve(prob),xx=sol.x

**结果如下：**



即最大收益为44千万