实验指导书： 线性规划模型

【实验目的】

1、通过上机采用MATLAB优化工具箱求解线性规划，进一步掌握有关线性规划问题的分析、建模与求解方法。

【实验相关知识】

一、线性规划问题的建模与求解

1．实验原理

（1）线性规划标准形式

$$\min_{x} z = f(x)$$
$$s.t.\quad g_i(x) \le 0 \qquad (i = 1,2,\cdots m)$$

其中目标函数 $f(x)$ 和约束条件中 $g_i(x)$ 都是线性函数

（2）MATLAB 软件求解线性规划的命令

1）命令格式 1

```
[x,fval]=linprog(c,A,b)
```

用于求解模型：

$$\min z = cX$$
$$s.t.\quad AX \le b$$

2）命令格式 2

```
[x,fval]=linprog(c,A,b,Aeq,beq)
```

用于求解模型：

$$\begin{aligned} &\min z = cX \\ &s.t.\quad AX \le b \\ &\qquad\quad AeqX = beq \end{aligned}$$

3）命令格式 3

```
[x,fval]=linprog(c,A,b,Aeq,beq,vlb,vub)
```

用于求解模型：

$$
\begin{aligned}
&\min z = cX \\
&s.t. \quad AX \le b \\
&\quad\quad AeqX = beq \\
&\quad\quad vlb \le X \le vub
\end{aligned}
$$

二、实验题目:

某工厂计划生产甲、乙两种产品，主要材料有钢材 3600 kg、铜材 2000 kg、专用设备能力 3000 台时。材料与设备能力的消耗定额以及单位产品所获利润如下表所示，问如何安排生产，才能使该厂所获利润最大。

材料 \ 消耗	甲（/件）	乙（/件）	现有材料与设备能力
钢材（kg)	9	4	3600
铜材（kg)	4	5	2000
设备能力（台时）	3	10	3000
利润（元）	70	120	

若用 10 元可以买到 1kg 铜材，问是否应该作这项投资？若投资，每天最多买多少 kg 铜材?

三、模型假设

解:

设甲、乙两种产品的计划产量分别为 x_1, x_2 件。生产这两种产品所消耗的钢材总数量为 $9x_1 + 4x_2$，但现在只有钢材 3600kg，因此，应有

$$
\begin{aligned}
&9x_1 + 4x_2 \le 3600 \\
&4x_1 + 5x_2 \le 2000 \\
&3x_1 + 10x_2 \le 3000 \\
&x_1 \ge 0, \quad x_2 \ge 0,
\end{aligned}
$$

总利润为

$$70x_1 + 120x_2 - 10 * m$$

LINGO 程序:

```
model:
max = 70*x1+120*x2;
9*x1+4*x2<=3600;
4*x1+5*x2<=2000;
3*x1+10*x2<=3000;
End
```

```
Global optimal solution found.
Objective value:                              42800.00
Infeasibilities:                              0.000000
Total solver iterations:                             2
```

Variable	Value	Reduced Cost
X1	200.0000	0.000000
X2	240.0000	0.000000

Row	Slack or Surplus	Dual Price
1	42800.00	1.000000
2	840.0000	0.000000
3	0.000000	13.60000
4	0.000000	5.200000

2、利用 matlab 求下面优化问题

运行程序

```
c=[-70 -120];
A=[9 4;4 5;3 10];
b=[3600;2000;3000];
Aeq=[];beq=[];
vlb=[0;0];vub=[];
[x,fval]=linprog(c,A,b,Aeq,beq,vlb,vub)
```

运行结果

```
Optimization terminated.
x =
   200.0000
   240.0000
fval =
-4.2800e+004
```

整数规划的 MATLAB 求解方法

（一） 用 MATLAB 求解整数规划问题

由于 MATLAB 优化工具箱中并未提供求解纯整数规划和混合整数规划的函数，因而需要自行根据需要和设定相关的算法来实现。现在有许多用户发布的工具箱可以解决该类问题。这里我们给出开罗大学的 Sherif 和 Tawfik 在 MATLAB Central 上发布的一个用于求解一般混合整数规划的程序，在此命名为 intprog，在原程序的基础上做了简单的修改，将其选择分枝变量的算法由自然序改造成分

枝变量选择原则中的一种，即：选择与整数值相差最大的非整数变量首先进行分枝。intprog 函数的调用格式如下：

[x,fval,exitflag]=intprog(c,A,b,Aeq,beq,lb,ub,M,TolXInteger)

该函数解决的整数规划问题为：

$$\begin{cases} \min \quad f = c^T x \\ s.t. \quad Ax \le b \\ \qquad A_{eq} x = b_{eq} \\ \qquad lb \le x \le ub \\ \qquad x_i \ge 0 \quad (i = 1,2,\cdots,n) \\ \qquad x_j \text{取整数（}j \in M\text{）} \end{cases}$$

在上述标准问题中，假设 x 为 n 维设计变量，且问题具有不等式约束 m_1 个，等式约束 m_2 个，那么：c、x 均为 n 维列向量，b 为 m_1 维列向量，b_{eq} 为 m_2 维列向量，A 为 $m_1 \times n$ 维矩阵，A_{eq} 为 $m_2 \times n$ 维矩阵。

在该函数中，输入参数有 c,A,b,A*eq*,b*eq*,lb,ub,M 和 TolXInteger。其中 c 为目标函数所对应设计变量的系数，A 为不等式约束条件方程组构成的系数矩阵，b 为不等式约束条件方程组右边的值构成的向量。Aeq 为等式约束方程组构成的系数矩阵，b*eq* 为等式约束条件方程组右边的值构成的向量。lb 和 ub 为设计变量对应的上界和下界。M 为具有整数约束条件限制的设计变量的序号，例如问题中设计变量为 $x_1, x_2, \cdots, x_6$，要求 x_2, x_3 和 x_6 为整数，则 M=[2;3;6]；若要求全为整数，则 M=1:6，或者 M=[1;2;3;4;5;6]。TolXInteger 为判定整数的误差限，即若某数 x 和最邻近整数相差小于该误差限，则认为 x 即为该整数。

在该函数中，输出参数有 x, fval 和 exitflag。其中 x 为整数规划问题的最优解向量，fval 为整数规划问题的目标函数在最优解向量 x 处的函数值，exitflag 为函数计算终止时的状态指示变量。

例 1　求解整数规划问题：

$$
\begin{cases}
\max & f = x_1 + x_2 \\
s.t. & 4x_1 - 2x_2 \geq 1 \\
 & 4x_1 + 2x_2 \leq 11 \\
 & 2x_2 \geq 1 \\
 & x_1, x_2 \geq 0，\text{且取整数值}
\end{cases}
$$

算法：

```
c=[-1;-1];
A=[-4 2;4 2;0 -2];
b=[-1;11;-1];
lb=[0;0];
M=[1;2];                         %均要求为整数变量
Tol=1e-8;                        %判断是否整数的误差限
[x,fval]=linprog(c,A,b,[],[],lb,[])           %求解原问题松弛线性规划
[x1,fval1]=intprog(c,A,b,[],[],lb,[],M,Tol)     %求最优解整数解
```

结果：

```
x =                              %松弛线性规划问题的最优解
    1.5000
    2.5000
fval =
   -4.0000
x1 =                             %整数规划的最优解
     2
     1
fval2 =
   -3.0000
```

（二） 用 **MATLAB** 求解 **0-1** 规划问题

在 MATLAB 优化工具箱中，提供了专门用于求解 0-1 规划问题的函数 bintprog，其算法基础即为分枝界定法，在 MATLAB 中调用 bintprog 函数求解 0-1 规划时，需要遵循 MATLAB 中对 0-1 规划标准性的要求。

0-1 规划问题的 MATLAB 标准型

$$\begin{cases} \min \quad f = c^T x \\ s.t. \quad Ax \le b \\ \qquad A_{eq} x = b_{eq} \\ \qquad x = 0,1 \end{cases}$$

在上述模型中，目标函数 f 需要极小化，以及需要满足的约束条件，不等式约束一定要化为形式为“≤”。

假设 x 为 n 维设计变量，且问题具有不等式约束 m_1 个，等式约束 m_2 个，那么：c、x 均为 n 维列向量，b 为 m_1 维列向量，b_{eq} 为 m_2 维列向量，A 为 $m_1 \times n$ 维矩阵，A_{eq} 为 $m_2 \times n$ 维矩阵。

如果不满足标准型的要求，则需要对原问题进行转化，化为标准型之后才能使用相关函数，标准化的方法和线性规划中的类似。

0-1 规划问题的 MATLAB 求解函数

MATLAB 优化工具箱中求解 0-1 规划问题的命令为 bintprog

bintprog 的调用格式

```
x = bintprog(f)
x = bintprog(f,A,b)
x = bintprog(f,A,b,Aeq,beq)
x = bintprog(f,A,b,Aeq,beq,x0)
x = bintprog(f,A,b,Aeq,Beq,x0,options)
[x,fval] = bintprog(...)
[x,fval,exitflag] = bintprog(...)
[x,fval,exitflag,output] = bintprog(...)
```

命令详解

1）x = bintprog(f)

该函数调用格式求解如下形式的 0-1 规划问题

$$\begin{cases} \min \quad f = c^T x \\ s.t. \quad x = 0,1 \end{cases}$$

2）x = bintprog(c,A,b)

该函数调用格式求解如下形式的 0-1 规划问题

$$\begin{cases} \min \quad f = c^T x \\ s.t. \quad Ax \le b \\ \qquad\quad x = 0,1 \end{cases}$$

3）x = bintprog (c,A,b,Aeq,beq)

该函数调用格式求解如下形式的 0-1 规划问题：

$$\begin{cases} \min \quad f = c^T x \\ s.t. \quad Ax \le b \\ \qquad\quad A_{eq}x = b_{eq} \\ \qquad\quad x = 0,1 \end{cases}$$

4）x = bintprog (c,A,b,Aeq,beq,x0)

该函数调用格式求解如下形式的 0-1 规划问题

$$\begin{cases} \min \quad f = c^T x \\ s.t. \quad Ax \le b \\ \qquad\quad A_{eq}x = b_{eq} \\ \qquad\quad x = 0,1 \end{cases}$$

在前一个调用格式的基础上同时设置求解算法的初始解为 x0，如果初始解 x0 不在 0-1 规划问题的可行域中，算法将采用默认的初始解

5）x = bintprog (c,A,b,Aeq,beq,x0,options)

用 options 指定的优化参数进行最小化。可以使用 optimset 来设置这些参数

上面的函数调用格式仅设置了最优解这一输出参数，如果需要更多的输出参数，则可以参照下面的调用格式：

[x,fval] = bintprog(...)

在优化计算结束之时返回整数规划问题在解 x 处的目标函数值 fval

[x,fval,exitflag] = bintprog(...)

在优化计算结束之时返回 exitflag 值，描述函数计算的退出条件。

[x,fval,exitflag,output] = bintprog(...)

在优化计算结束之时返回结构变量 output

例 2：求解 0-1 规划问题

$$
\begin{cases}
\max \quad f=\sum_{i=1}^{n}\sum_{j=1}^{n}E_{ij}x_{ij} \\
s.t. \quad \sum_{j=1}^{n}x_{ij}=1 \quad (i=1,2,\cdots,n) \\
\qquad \sum_{i=1}^{n}x_{ij}=1 \quad (j=1,2,\cdots,n) \\
\qquad x_{ij}=0\text{或}1\text{（1,2,3,4）}(i=1,2,\cdots,n;j=1,2,\cdots,n)
\end{cases}
\qquad
E=\begin{bmatrix} 20 & 12 & 33 & 26 \\ 22 & 15 & 29 & 23 \\ 21 & 13 & 31 & 24 \\ 22 & 16 & 32 & 23 \end{bmatrix}
$$

为了程序的可读性，我们用一维下标来表示设计变量，即用$x_1 \sim x_4$表示$x_{11} \sim x_{14}$，用$x_5 \sim x_8$表示$x_{21} \sim x_{24}$，用$x_9 \sim x_{12}$表示$x_{31} \sim x_{34}$，用$x_{13} \sim x_{16}$表示$x_{41} \sim x_{44}$，于是约束条件和目标函数分别为：

$$
\begin{cases}
x_1+x_2+x_3+x_4=1 \\
x_5+x_6+x_7+x_8=1 \\
x_9+x_{10}+x_{11}+x_{12}=1 \\
x_{13}+x_{14}+x_{15}+x_{16}=1 \\
x_1+x_5+x_9+x_{13}=1 \\
x_2+x_6+x_{10}+x_{14}=1 \\
x_3+x_7+x_{11}+x_{15}=1 \\
x_4+x_8+x_{12}+x_{16}=1 \\
x_i=0,1 \quad (i=1,2,\cdots,16)
\end{cases}
$$

$$f=E_{11}x_1+E_{12}x_2+E_{13}x_3+E_{14}x_4+E_{21}x_5+E_{22}x_6+\cdots+E_{44}x_{16}$$

算法：

```
c=[20;12;33;26;22;15;29;23;21;13;31;24;22;16;32;23];
Aeq=[1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0;
     0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0;
     0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0;
     0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1;
     1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0;
     0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0;
     0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0;
     0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1;
     ];
```

beq=ones(8,1);

vlb=zeros(1,16);

vub=ones(1,16);

[x,fval]=linprog(c,[],[],Aeq,beq,vlb,vub)

B=reshape(x,4,4); %由于 x 是一列元素，为了使结果更加直观，故排成与效率矩阵 E 相对应的形式

B'

fval fval 结果：

ans =

0.4674	0.5326	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000
0.5326	0.4674	0.0000	0.0000
0.0000	0.0000	0.0000	1.0000

fval =

85.0000

方法 2：利用 intlinprog 命令

```
c=[20;12;33;26;22;15;29;23;21;13;31;24;22;16;32;23];

Aeq=[1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0;

     0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0;

     0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0;

     0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1;

     1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0;

     0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0;

     0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0;

     0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1;

     ];

beq=ones(8,1);

intcon=1:16;

vlb=zeros(1,16);
```

```
vub=ones(1,16);
[x,fval]=intlinprog(c,intcon,[],[],Aeq,beq,vlb,vub)

B=reshape(x,4,4); %由于 x 是一列元素，为了使结果更加直观，故排成与效率矩阵 E 相对应的形式
B'
Fval
```

运行结果：

LP: Optimal objective value is 85.000000.

Optimal solution found.

Intlinprog stopped at the root node because the objective value is within a gap tolerance of the optimal value,

options.TolGapAbs = 0 (the default value). The intcon variables are integer within tolerance,

options.TolInteger = 1e-05 (the default value).

ans =

```
     1     0     0     0
     0     0     1     0
     0     1     0     0
     0     0     0     1
```

fval =

```
    85
```

整数规划的应用——组件配套问题

某机械产品需要由三个工厂开工一起生产后组装完成。每件机械需要 4 个组件 1 和 3 个组件 2。生产这两种组件需要消耗两种原材料 A 和 B。已知这两种原材料的供应量分别为 400kg 和 600kg。

由于三个工厂的生产条件和拥有设备工艺条件不同，每个工厂生产组件的能力和原材料的消耗也不尽相同，且每个工厂开工一次都是配套生产一定数量的组件 1 和组件 2，其具体的数据如表所示。

表 11-2 各工厂生产能力和消耗原材料的数据表

	每个工厂消耗原材料的数量（公斤）		每个工厂各组件的生产能力（件数）	
	A 材料	B 材料	组件 1	组件 2
工厂 1	9	7	8	6
工厂 2	6	10	7	9
工厂 3	4	9	9	5

现在的最优化问题是，这三个工厂应当如何安排生产，才能使该产品的配套数达到最大？

（Ⅰ）组件配套问题的建模

设 x_1、x_2 和 x_3 是三个工厂分别开工的次数，将其作为该问题的设计变量。由于每个工厂开工一次都是配套生产，故每次开工消耗的原材料一定，且生产的组件数也是一定的。在这个假设的前提之下，我们可以得出该问题的目标函数和约束条件。

因为原材料的总量是有限的，根据工厂的开工次数，可得工厂 1 将消耗 A 材料 $9x_1$，工厂 2 将消耗 A 材料 $6x_2$，工厂 3 将消耗 A 材料 $4x_3$，故有约束条件：

$$9x_1 + 6x_2 + 4x_3 \le 400$$

同理，对于材料 B 的总量约束条件可以表达为：$7x_1 + 10x_2 + 9x_3 \le 600$

然后再来分析三个工厂零件生产的情况，三个工厂生产的组件 1 的数量分别为 $8x_1$、$7x_2$ 和 $9x_3$，故组件 1 的总数为：$Q_1 = 8x_1 + 7x_2 + 9x_3$

同理，组件 2 的总数为：$Q_2 = 6x_1 + 9x_2 + 5x_3$

下一步是分析目标函数，该问题要求的不是生产的各种组件总数最多，而是要求产品的配套数量最大，即能组成的机械数目最多。问题中已经给出了该种机械中两种组件的配比为 4:3，故组件 1 所能成套的数目 T_1 满足约束条件：

$$T_1 \le \frac{Q_1}{4} = \frac{8x_1 + 7x_2 + 9x_3}{4}$$

同理，组件 2 所能成套的数目 T_2 满足约束条件：$T_2 \le \dfrac{Q_2}{3} = \dfrac{6x_1 + 9x_2 + 5x_3}{3}$

因而，所能组成的成品机械的数目 f 应该为 T_1 和 T_2 中的较小值，即：

$f = \min(T_1, T_2)$

那么，我们求解的目标即是使得 f 能够尽可能大，可以看出，这种形式并不是有关设计变量的线性函数，我们需要对其进行转化，此时我们可以令一个人工设计变量等于目标函数值，则有：$x_4 = \min(T_1, T_2)$

在该假设下，一定满足关系式：$T_1 \ge x_4$ 且 $T_2 \ge x_4$

结合约束关系，对 T_1 的约束可以表示为：$x_4 \le T_1 \le \dfrac{Q_1}{4} = \dfrac{8x_1 + 7x_2 + 9x_3}{4}$

同理，对 T_2 的约束可以表示为：$x_4 \le T_2 \le \dfrac{Q_2}{3} = \dfrac{6x_1 + 9x_2 + 5x_3}{3}$

对 T_1 的上述关系进行整理，可以得到关系：$-8x_1 - 7x_2 - 9x_3 + 4x_4 \le 0$

同理对 T_2 也可以得到不等式关系为：$-6x_1 - 9x_2 - 5x_3 + 3x_4 \le 0$

上面两个式子即为由组件的配比数得到的关于组件数目的约束条件，此时问题的目标就是如何使得 x_4 取到最大值，由于开工的次数一定是一个非负整数，故也是一个约束条件，决定了该问题是一个纯整数规划问题。结合前面给出的原材料约束，可以得到如下的数学模型：

$$\begin{cases} \max \quad f = x_4 \\ s.t. \quad 9x_1 + 6x_2 + 4x_3 \le 400 \\ \qquad 7x_1 + 10x_2 + 9x_3 \le 600 \\ \qquad -8x_1 - 7x_2 - 9x_3 + 4x_4 \le 0 \\ \qquad -6x_1 - 9x_2 - 5x_3 + 3x_4 \le 0 \\ \qquad x_i \ge 0 \text{且取整数值} (i = 1,2,3,4) \end{cases}$$

（II）组件配套问题的求解

利用§8. 1 节中给出的ＭＡＴＬＡＢ函数对此问题求解，代码和运行结果如下：

算法：

```
%目标函数所对应的设计变量的系数，为转化为极小，故取原目标函数的相反数
f=[0;0;0;-1];
%不等式约束
```

```
A=[ 9    6    4 0;
     7 10    9 0;
   -8 -7 -9 4;
   -6 -9 -5 3];
b=[400;600;0;0];
%边界约束，由于无上界，故设置 ub=[Inf;Inf;Inf;Inf];
lb=[0;0;0;0];
ub=[Inf;Inf;Inf;Inf];
%所有变量均为整数变量，故将所有序号组成向量 M
M=[1;2;3;4];
%判定为整数的误差限
Tol=1e-8;
%求最优解 x 和目标函数值 fval，并返回状态指示
[x,fval,exitflag]=intprog(f,A,b,[],[],lb,ub,M,Tol)
```

结果：

```
x =                              %最优解向量 x
      18
      15
      36
      141
fval =                           %在最优解向量 x 处，原目标函数值的相反数
  -141.000
exitflag=                        %算法终止指示变量，说明问题收敛到了最优解 x
      1
```

由运行结果可知，工厂 1 、2 和 3 需要分别开工 18、15 和 36 次，这样所能生产出来的成套的机械为 141 件。

Solve the problem

$$\min_x (-3x_1 - 2x_2 - x_3) \text{ subject to } \begin{cases} x_3 \text{ binary} \\ x_1, x_2 \geq 0 \\ x_1 + x_2 + x_3 \leq 7 \\ 4x_1 + 2x_2 + x_3 = 12 \end{cases}$$

without showing iterative display.

Specify the solver inputs.

```
f = [-3;-2;-1];
intcon = 3;
A = [1,1,1];
b = 7;
Aeq = [4,2,1];
beq = 12;
lb = zeros(3,1);
ub = [Inf;Inf;1]; % enforces x(3) is binary
```

Specify no display.

```
options = optimoptions('intlinprog','Display','off');
```

Run the solver.

```
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,options)
```

```
x =

         0
    5.5000
    1.0000
```