到最大.

建模步骤：

（1）寻求决策.

该问题需要回答三个车间各应开多少班，如何判定产品的配套数、配套数的最大值是多少等问题.

（2）确定决策变量.

设甲、乙、丙三个车间所开的生产班数分别是 x_1, x_2, x_3.

（3）确定优化目标.

该问题的目标函数是要使产品的配套数最大，甲、乙、丙生产 A 零件总数是 $7x_1+6x_2+8x_3$，生产 B 零件总数是 $5x_1+9x_2+4x_3$. 而每件产品要 4 个 A 零件，3 个 B 零件，所以产品的最大量不超过 $\frac{7x_1+6x_2+8x_3}{4}$ 和 $\frac{5x_1+9x_2+4x_3}{3}$ 中较小的一个. 因此,产品的配套数 $S=\min\left\{\frac{7x_1+6x_2+8x_3}{4},\frac{5x_1+9x_2+4x_3}{3}\right\}$.

（4）寻找约束条件.

由于原材料的总量有限，所以三个车间所用原料 1 和原料 2 的总和应分别小于 300 公斤和 500 公斤，即 $8x_1+5x_2+3x_3\leqslant 300$，$6x_1+9x_2+8x_3\leqslant 500$. 另外，根据问题的实际情况，还有决策变量的非负约束.

（5）构成数学模型.

将目标函数和约束条件放在一起，即得到数学模型：

$$\max \quad S=\min\left\{\frac{7x_1+6x_2+8x_3}{4},\frac{5x_1+9x_2+4x_3}{3}\right\}$$
$$\text{s.t.}\quad \begin{cases}8x_1+5x_2+3x_3\leqslant 300\\ 6x_1+9x_2+8x_3\leqslant 500\\ x_1,x_2,x_3\geqslant 0\end{cases} \tag{1.7}$$

模型转化：模型（1.7）的目标函数不是线性函数，但可以通过适当的变换化为线性函数，设

$$y=\min\left\{\frac{7x_1+6x_2+8x_3}{4},\frac{5x_1+9x_2+4x_3}{3}\right\}$$

则上式可以等价于下面两个不等式：

$$\frac{7x_1+6x_2+8x_3}{4}\geqslant y\ ,\ \frac{5x_1+9x_2+4x_3}{3}\geqslant y \tag{1.8}$$

将条件（1.8）化为线性条件后，模型（1.7）可转化为

$$\max \quad S = y$$

$$\text{s.t.} \quad \begin{cases} 7x_1 + 6x_2 + 8x_3 - 4y \geqslant 0 \\ 5x_1 + 9x_2 + 4x_3 - 3y \geqslant 0 \\ 8x_1 + 5x_2 + 3x_3 \leqslant 300 \\ 6x_1 + 9x_2 + 8x_3 \leqslant 500 \\ x_1, x_2, x_3, y \geqslant 0 \end{cases} \tag{1.9}$$

模型求解：将模型（1.9）化为如下标准形式：

$$\min \quad S' = -y$$

$$\text{s.t.} \quad \begin{cases} -7x_1 - 6x_2 - 8x_3 + 4y \leqslant 0 \\ -5x_1 - 9x_2 - 4x_3 + 3y \leqslant 0 \\ 8x_1 + 5x_2 + 3x_3 \leqslant 300 \\ 6x_1 + 9x_2 + 8x_3 \leqslant 500 \\ x_1, x_2, x_3, y \geqslant 0 \end{cases} \tag{1.10}$$

（1）Matlab 软件求解.

令 $X = (x_1, x_2, x_3, y)^{\mathrm{T}}$，根据模型（1.10）中的目标函数和约条件的系数，在 Matlab 指令窗口输入求解程序:

```
c=[0,0,0,-1];
A=[-7,-6,-8,4;-5,-9,-4,3;8,5,3,0;6,9,8,0];
b=[0,0,300,500]';
Aeq=[];
beq=[];
lb=zeros(4,1);
ub=[inf,inf,inf,inf];
[X,fval]=linprog(c,A,b,Aeq,beq,lb,ub);
S=-fval;
X,S
```

回车后，得到结果为：

```
X =
    15. 1232
    15. 7179
    33. 4749
   116. 9924
S =
   116. 9924
```

（2）Lingo 软件求解.

利用 Lingo 软件求解优化模型时，不需要像 Matlab 那样将模型（1.9）化为标准形（1.10）后才能进行. 可以直接在 Lingo 模型窗口中按模型（1.9）输入：

```
max=y;
7*x1+6*x2+8*x3-4*y>=0;
5*x1+9*x2+4*x3-3*y>=0;
8*x1+5*x2+3*x3<=300;
6*x1+9*x2+8*x3<=500;
```

求解结果如下：

```
Global optimal solution found.
    Objective value:                        116. 9924
    Total solver iterations:                       4

                 Variable        Value        Reduced Cost
                        Y      116. 9924          0. 000000
                       X1      15. 12319          0. 000000
                       X2      15. 71793          0. 000000
                       X3      33. 47494          0. 000000

                      Row    Slack or Surplus     Dual Price
                        1      116. 9924          1. 000000
                        2      0. 000000         -0. 1537808
                        3      0. 000000         -0. 1282923
                        4      0. 000000          0. 7136788E-01
                        5      0. 000000          0. 1911640
```

两种软件的求解结果显示，当甲、乙、丙三个车间所开的生产班数分别为 15.123 2、15.717 9 和 33.474 9 时，生产的产品配套数达到最大，为 116.992 4 套.

注记：一般而言，车间所开的生产班数和产品最大配套数应为整数. 因此，该线性规划模型的决策变量还应为整数. 在增加这一约束条件后，又该如何求解该模型呢？我们将在下一节介绍整数规划模型后给出其解法.

1.2 整数规划模型

当模型（1.1）中的决策变量 x_i $(i=1,2,\cdots,n)$（部分或全部）均为整数时称模型（1.1）为整数规划模型. 整数规划模型可以是线性的，也可以是非线性的. 整数规划模型大致可分为三类：当变量全限制为整数时，称之为纯（完全）整数规划模型；当变量部分限制为整数时，称之为混合整数规划模型；当变量只能取 0 或 1 时，称之为 0-1 规划模型. 当整数规划模型为线性规划模型时，称之为整数线性规划模型.

在整数线性规划模型中，为了满足变量为整数的要求，初看起来似乎只要把已得的非整数解去掉非整数部分化整就可以了，实际上化整后得到的解不一定是可行解和最优解. 若原

线性规划最优解全是整数，则整数线性规划最优解与线性规划最优解一致，此时可以用求解线性规划模型的方法求解整数线性规划模型. 然而，大多数整数规划模型，需要寻求特殊的解法. 由于至今尚未找到一般的求解整数规划模型多项式解法，本节仅针对一些经典的整数规划问题，讨论建模过程和相应的计算机算法.

1.2.1 整数规划的计算机解法

对于一般的整数规划模型，无法直接利用 Matlab 的函数求解，但可以使用 Lingo 等专用软件求解.

针对上一节例 1.2 中给出的模型（1.9），增加 x_1 ，x_2 ，x_3 和 y 为整数的约束后，该模型即整数线性规划模型. 利用 Lingo 软件求解 ,只需在例 1.2 求解的命令中增加变量的整数限制即可. 因此，直接在 Lingo 模型窗口中输入：

```
max=y;
7*x1+6*x2+8*x3-4*y>=0;
5*x1+9*x2+4*x3-3*y>=0;
8*x1+5*x2+3*x3<=300;
6*x1+9*x2+8*x3<=500;
@gin(x1);@gin(x2);@gin(x3);@gin(y);
```

求解结果如下：

```
Global optimal solution found.
    Objective value:                          116. 0000
    Extended solver steps:                        0
    Total solver iterations:                      6

                    Variable        Value        Reduced Cost
                           Y     116. 0000         -1. 000000
                          X1     15. 00000          0. 000000
                          X2     16. 00000          0. 000000
                          X3     33. 00000          0. 000000
                         Row   Slack or Surplus     Dual Price
                           1     116. 0000          1. 000000
                           2     1. 000000          0. 000000
                           3     3. 000000          0. 000000
                           4     1. 000000          0. 000000
                           5     2. 000000          0. 000000
```

求解结果显示，当甲、乙、丙三个车间所开的生产班数分别为 15、16 和 33 时，生产的产品配套数达到最大，为 116 套.

注记：若将例 1.2 中模型的求解结果 $x_1=15.1232$、$x_2=15.7179$、$x_3=33.4749$、$y=116.9924$ 按四舍五入的方法化整作为该整数线性规划模型的解，则 $x_1=15$、$x_2=14$、$x_3=33$、$y=117$. 此时，$7x_1+6x_2+8x_3-4y=-3$，不满足模型（1.9）中 $7x_1+6x_2+8x_3-4y\geqslant 0$ 的约束条件. 因此，在整数线性规划模型中，通过把已得的非整数解舍入化整来得到可行解或最优解，一般是不可行的.

1.2.2　0-1 整数规划

在部分规划问题中，每个需要做的决策只有两种时，可以使用 0-1 整数规划来建模. 而 0-1 整数规划是整数规划中的特殊情形，它的变量 x_i 仅取值 0 或 1. 对于 0-1 整数规划模型，除了可以使用 Lingo 软件求解外，也可以利用 Matlab 进行求解. Matlab 中规定 0-1 规划模型的标准形式为：

$$\min \quad \boldsymbol{cx}$$
$$\text{s.t.} \quad \begin{cases} \boldsymbol{Ax}\leqslant \boldsymbol{b} \\ \mathrm{Aeq}\cdot\boldsymbol{x}=\mathrm{beq} \end{cases}$$

其中，$\boldsymbol{x}$ 的每一个分量的取值为 0 或 1. Matlab 中求解 0-1 规划的命令为：

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

其中，x 返回的是决策向量 $\boldsymbol{x}$ 的取值，fval 返回的是目标函数的最优值. 然而，使用 Matlab 软件求解数学规划模型有一个缺陷，即必须把所有的决策变量化成一维决策变量. 实际上对于很多多维变量的数学规划模型，尽管通过变量替换后就能化为一维决策变量，但约束条件会变得很难表示.

例 1.3　指派问题

问题：拟分配 n 人去干 n 项工作，每人干且仅干一项工作，若分配第 i 人去干第 j 项工作，需花费 c_{ij} 单位时间，问应如何分配工作才能使工人花费的总时间最少.

问题分析：在该问题中，每一个人均能从事所有的工作，只是从事不同的工作需要花费的单位时间不同而已. 由于每人干且仅干一项工作，所以对第 i 人和第 j 项工作之间，只存在干与不干的关系. 因此，在第 i 人和第 j 项工作之间可以分别用 1 和 0 来刻画干与不干，进而建立 0-1 规划模型.

建模步骤：

（1）寻求决策.

该问题需要决策的是第 i 人和第 j 项工作之间的干与不干关系，以达到使工人花费的总时间最少的目的.

（2）确定决策变量.

由于每个人和每项工作之间均需建立关系，所以可用变量 x_{ij} 来描述第 i 人和第 j 项工作之间的关系. 若分配第 i 人去干第 j 项工作，则取 $x_{ij}=1$，否则取 $x_{ij}=0$.

（3）确定优化目标.

该问题的目标是使工人花费的总时间最少. 对第 i 人而言，若被分配去干第 j 项工作（此时 $x_{ij}=1$），则需花费的单位时间为 c_{ij}；若不被分配去干第 j 项工作（此时 $x_{ij}=0$），则需花费的单位时间为 0. 因此，第 i 人花费在第 j 项工作的时间可用 $c_{ij}x_{ij}$ 表示. 由于在所有的工作中，第 i 人干且仅干一项工作，若第 i 人被分配去干第 j_0 项工作，则当 $j \neq j_0$ 时，$c_{ij}x_{ij}=0$，所以第 i 人总共花费的单位时间为 $c_{ij_0}x_{ij_0}=\sum_{j=1}^{n}c_{ij}x_{ij}$. 因此，在该问题中工人花费的总时间为 $T=\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij}$.

（4）寻找约束条件.

根据每人干且仅干一项工作的要求，对第 i 人而言，应有 $\sum_{j=1}^{n}x_{ij}=1$；对第 j 项工作而言，应有 $\sum_{i=1}^{n}x_{ij}=1$.

（5）构成数学模型.

将目标函数和约束条件放在一起，即得上述指派问题的数学模型：

$$
\begin{aligned}
&\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}x_{ij}\\
&\text{s.t.} \quad \begin{cases}
\sum_{j=1}^{n}x_{ij}=1 \quad (i=1,2,\cdots,n)\\
\sum_{i=1}^{n}x_{ij}=1 \quad (j=1,2,\cdots,n)\\
x_{ij}=0 \text{ 或 } 1 \quad (i,j=1,2,\cdots,n)
\end{cases}
\end{aligned}
\tag{1.11}
$$

容易看出，要给出一个指派问题的实例，只需给出矩阵 $\boldsymbol{C}=(c_{ij})$，$\boldsymbol{C}$ 被称为指派问题的系数矩阵. 下面针对具体系数矩阵

$$
\boldsymbol{C}=\begin{pmatrix}
3 & 8 & 2 & 10 & 3\\
8 & 7 & 2 & 9 & 7\\
6 & 4 & 2 & 7 & 5\\
8 & 4 & 2 & 3 & 5\\
9 & 10 & 6 & 9 & 10
\end{pmatrix}
$$

给出计算机解法.

模型求解：

（1）Matlab 软件求解.

由于 $x_{ij}\ (i,j=1,\cdots,5)$ 为二维决策变量，所以利用 Matlab 求解需要将其变为一维决策变量 $y_k\ (k=1,\cdots,25)$，在 Matlab 指令窗口中输入求解程序：

```
c=[3 8 2 10 3;8 7 2 9 7;6 4 2 7 5;8 4 2 3 5;9 10 6 9 10];
c=c(:);
a=zeros(10,25);
for i=1:5
    a(i,(i-1)*5+1:5*i)=1;
    a(5+i,i:5:25)=1;
end
b=ones(10,1);
[y,fval]=bintprog(c,[],[],a,b);
x=reshape(y,5,5),fval
%x=reshape(y,5,5) 表示将 y 转化为矩阵 5*5 的矩阵 x,x 中元素按列从 y 中抽取.
```

回车后，得到结果为：

```
x =
        0     0     0     0     1
        0     0     1     0     0
        0     1     0     0     0
        0     0     0     1     0
        1     0     0     0     0
fval =
      21
```

（2）Lingo 软件求解.

在 Lingo 模型窗口中直接输入：

```
model:
sets:
var/1. . 5/;
link(var,var):c,x;
endsets
data:
c=3 8 2 10 3
   8 7 2 9 7
   6 4 2 7 5
   8 4 2 3 5
   9 10 6 9 10;
enddata
min=@sum(link:c*x);
@for(var(i):@sum(var(j):x(i,j))=1);
@for(var(j):@sum(var(i):x(i,j))=1);
@for(link:@bin(x));
```

```
end
```

求解输出(只列需要的结果):

```
Global optimal solution found.
    Objective value:                          21. 00000
    Extended solver steps:                        0
    Total solver iterations:                      0

                Variable        Value         Reduced Cost
                X( 1,1)      0. 000000          3. 000000
                X( 1,2)      0. 000000          8. 000000
                X( 1,3)      0. 000000          2. 000000
                X( 1,4)      0. 000000          10. 00000
                X( 1,5)      1. 000000          3. 000000
                X( 2,1)      0. 000000          8. 000000
                X( 2,2)      0. 000000          7. 000000
                X( 2,3)      1. 000000          2. 000000
                X( 2,4)      0. 000000          9. 000000
                X( 2,5)      0. 000000          7. 000000
                X( 3,1)      0. 000000          6. 000000
                X( 3,2)      1. 000000          4. 000000
                X( 3,3)      0. 000000          2. 000000
                X( 3,4)      0. 000000          7. 000000
                X( 3,5)      0. 000000          5. 000000
                X( 4,1)      0. 000000          8. 000000
                X( 4,2)      0. 000000          4. 000000
                X( 4,3)      0. 000000          2. 000000
                X( 4,4)      1. 000000          3. 000000
                X( 4,5)      0. 000000          5. 000000
                X( 5,1)      1. 000000          9. 000000
                X( 5,2)      0. 000000          10. 00000
                X( 5,3)      0. 000000          6. 000000
                X( 5,4)      0. 000000          9. 000000
                X( 5,5)      0. 000000          10. 00000
```

求解结果显示,当分配第 1 人去完成第 5 项工作、第 2 人去完成第 3 项工作、第 3 人去完成第 2 项工作、第 4 人去完成第 4 项工作、第 5 人去完成第 1 项工作时,工人花费的单位时间最少,为 21.

例 1.4 某公司有 5 个项目被列入投资计划,各项目的投资额和期望的投资收益如表 1.3

所示，该公司只有 600 百万资金可用于投资，由于技术上的原因投资受到以下约束：

（1）在项目 1，2 和 3 中必须有一项被选中；

（2）项目 3 和 4 只能选中一项；

（3）项目 5 被选中的前提是项目 1 必须被选中，

如何在上述条件下选择一个最好的投资方案，使投资收益最大？

表 1.3　各项目的投资额和投资收益情况表

项目	投资额（百万元）	投资收益（百万元）
1	210	150
2	300	210
3	100	60
4	130	80
5	260	180

问题分析：该问题是投资决策问题. 由于每个项目的投资额与投资收益是固定的，因此对每一个项目而言，不存在投资数量多少的问题，只存在是否被选中投资的问题. 针对这种两者择一的问题，可以通过引入 0-1 变量建立 0-1 规划模型.

建模步骤：

（1）寻求决策.

该问题是在已有资金和技术的条件下，需要决策的是第 i 个项目是否被选中投资，建立最好的投资方案，使得投资收益最大.

（2）确定决策变量.

由于每个项目只存在是否被选中投资的问题，所以可用 x_i 来描述第 i 个项目是否被选中投资. 若第 i 个项目被选中投资，则取 $x_i = 1$，否则取 $x_i = 0$.

（3）确定优化目标.

该问题的目标是使投资收益最大. 根据决策变量的取值，对第 i 个项目而言，无论是否被选中投资，其投资收益均可用其对应收益额与 x_i 的乘积表示. 因此，该问题的投资收益总额为 $S = 150x_1 + 210x_2 + 60x_3 + 80x_4 + 180x_5$.

（4）寻找约束条件.

由于该公司只有 600 百万资金可用于投资，所以投资不能超过公司的 600 百万资金，故有条件 $210x_1 + 300x_2 + 100x_3 + 130x_4 + 260x_5 \leqslant 600$. 在项目 1，2 和 3 中必须有一项被选中，故有条件 $x_1 + x_2 + x_3 = 1$，项目 3 和 4 只能选中一项，故有条件 $x_3 + x_4 \leqslant 1$，项目 5 被选中的前提是项目 1 必须被选中，故有条件 $x_5 \leqslant x_1$，即 $-x_1 + x_5 \leqslant 0$.

（5）构成数学模型.

将目标函数和约束条件放在一起，即得该投资问题的 0-1 规划数学模型：

$$\max S = 150x_1 + 210x_2 + 60x_3 + 80x_4 + 180x_5$$
$$\text{s.t.} \begin{cases} 210x_1 + 300x_2 + 100x_3 + 130x_4 + 260x_5 \leqslant 600 \\ x_1 + x_2 + x_3 = 1 \\ x_3 + x_4 \leqslant 1 \\ -x_1 + x_5 \leqslant 0 \\ x_i = 0\text{或}1 \ \ (i = 1, 2\cdots, 5) \end{cases} \tag{1.12}$$

模型求解：

（1）Matlab 软件求解.

在 Matlab 指令窗口中输入求解程序：

```
c=-[150,210,60,80,180];   %转换求 max 为求 min
a=[210,300,100,130,260;0,0,1,1,0;-1,0,0,0,1];
b=[600;1;0];
Aeq=[1,1,1,0,0];
beq=1;
[y,fval]=bintprog(c,a,b,Aeq,beq);
x=y';S=-fval;
x,S
```

回车后，得到结果为：

```
x =
        1         0         0         1         1
S =
    410
```

（2）Lingo 软件求解.

在 Lingo 模型窗口中直接输入程序：

```
max=150*x1+210*x2+60*x3+80*x4+180*x5;
210*x1+300*x2+100*x3+130*x4+260*x5<600;
x1+x2+x3=1;
x3+x4<1;
-x1+x5<0;
@bin(x1);@bin(x2);@bin(x3);@bin(x4);@bin(x5);
```

求解输出：

```
Global optimal solution found.
    Objective value:                                  410. 0000
    Extended solver steps:                                    0
    Total solver iterations:                              0
```

Variable	Value	Reduced Cost
X1	1. 000000	-150. 0000
X2	0. 000000	-210. 0000
X3	0. 000000	-60. 00000
X4	1. 000000	-80. 00000
X5	1. 000000	-180. 0000

Row	Slack or Surplus	Dual Price
1	410. 0000	1. 000000
2	0. 000000	0. 000000
3	0. 000000	0. 000000
4	0. 000000	0. 000000
5	0. 000000	0. 000000

求解结果表明：该公司选择投资项目 1、4、5 时，预期投资收益最大，为 410 百万元.

1.3　非线性规划模型

目标函数或约束条件中至少有一个是非线性函数的数学规划模型，称为非线性规划模型. 非线性规划模型一般可写为：

$$\begin{aligned}&\min(\max)\quad z=f(\boldsymbol{x})\\&\text{s.t.}\ \begin{cases}h_j(\boldsymbol{x})\leqslant 0 & (j=1,\cdots,q)\\ g_i(\boldsymbol{x})=0 & (i=1,\cdots,p)\end{cases}\end{aligned}\tag{1.13}$$

其中，$\boldsymbol{x}=[x_1\ \ \cdots\ \ x_n]^{\mathrm{T}}$ 称为决策变量，$f(\boldsymbol{x})$ 称为目标函数，$g_i(\boldsymbol{x})$ $(i=1,\cdots,p)$ 和 $h_j(\boldsymbol{x})$ $(j=1,\cdots,q)$ 称为约束函数. 另外，$g_i(\boldsymbol{x})=0$ $(i=1,\cdots,p)$ 称为等式约束，$h_j(\boldsymbol{x})\leqslant 0$ $(j=1,\cdots,q)$ 称为不等式约束.

若模型（1.13）中无约束条件，则称其为无约束极值问题，即：

$$\min(\max)\ z=f(\boldsymbol{x}),\quad \boldsymbol{x}\in\mathbf{R}^n\tag{1.14}$$

若模型（1.13）的目标函数为自变量 $\boldsymbol{x}$ 的二次函数，约束条件又全是线性的，就称其为二次规划模型. 二次规划模型的标准形式可表述如下：

$$\begin{aligned}&\min\ \ z=\frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}+\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}\\&\text{s.t.}\ \begin{cases}\boldsymbol{A}\boldsymbol{x}\leqslant\boldsymbol{b}\\ \mathrm{Aeq}\cdot\boldsymbol{x}=\mathrm{beq}\\ \mathrm{lb}\leqslant\boldsymbol{x}\leqslant\mathrm{ub}\end{cases}\end{aligned}\tag{1.15}$$

这里 $\boldsymbol{H}$ 是实对称矩阵，$\boldsymbol{c},\boldsymbol{b},\mathrm{beq},\mathrm{lb},\mathrm{ub}$ 是列向量，$\boldsymbol{A},\mathrm{Aeq}$ 是相应维数的矩阵.

一般来说，求解非线性规划问题要比求解线性规划问题困难得多. 线性规划模型如果有最优解，其最优解必然在可行域的顶点（或边界）上取得，而非线性规划模型的最优解可能在可行域的任一点上取得. 目前，还没有适于各种非线性规划问题的一般算法，已有的算法都有自己特定的适用范围，带有一定的局限性. 因此，为了求解带有约束条件的非线性规划模型，常常通过将约束问题化为无约束问题、将非线性规划问题化为线性规划问题、将复杂问题化为较简单的问题等途径，进而用已有的算法求解.

本节主要介绍无约束优化、二次规划和非线性规划的 Matlab 求解命令，并通过实例分析非线性规划模型的建立和求解.

1.3.1 无约束极值问题

1.3.1.1 一元函数无约束极值问题

对于一元函数无约束优化问题：

$$\min\ f(\boldsymbol{x}),\ \ a \leqslant x \leqslant b$$

Matlab 中的求解命令为：

```
[x,fval]=fminbnd('fun',a,b)
```

其中，fun 是函数 $f(x)$ 的 M 文件，x 返回的是决策变量 $\boldsymbol{x}$ 的取值，fval 返回的是目标函数的最优值.

例 1.5 有边长为 3m 的正方形铁板，在四个角剪去相等的正方形以制成方形无盖水槽，问如何剪使水槽的容积最大？

建立模型：设剪去的正方形的边长为 x，则水槽的容积为 $(3-2x)^2 x$，从而建立无约束优化模型为 $\max\ \ (3-2x)^2 x,\quad 0<x<1.5$.

模型求解：

先编写 M 文件：

```
function f=fun1_5(x)
f=-(3-2*x). ^2*x;% 转换求 max 为求 min
```

再在 Matlab 运行窗口中输入：

```
[x,fval]=fminbnd('fun1_5',0,1. 5);
xmax=x
fmax=-fval
```

运行结果为：

```
xmax =
      0. 5000
fmax =
```