数学建模算法与应用

第3章 非线性规划

3.1 非线性规划模型

3.1.1 非线性规划的实例与定义

如果目标函数或约束条件中包含非线性函数，就称这种规划问题为非线性规划问题。一般说来，解非线性规划要比解线性规划问题困难得多。而且，也不像线性规划有单纯形法这一通用方法，非线性规划目前还没有适于各种问题的一般算法，各个方法都有自己特定的适用范围。

例 3.1 （投资决策问题）某企业有n个项目可供选择投资，并且至少要对其中一个项目投资。已知该企业拥有总资金A元，投资于第$i(i=1,\cdots,n)$个项目需花资金a_i元并预计可收益b_i元。试选择最佳投资方案。

解 设投资决策变量为

$$x_i=\begin{cases}1, & \text{决定投资第}i\text{个项目}\\ 0, & \text{决定不投资第}i\text{个项目}\end{cases},\quad i=1,\cdots,n,$$

则投资总额为$\sum_{i=1}^{n}a_i x_i$，投资总收益为$\sum_{i=1}^{n}b_i x_i$。

因为该公司至少要对一个项目投资，并且总的投资金额不能超过总资金A，故有限制条件

$$0 < \sum_{i=1}^{n} a_i x_i \le A,$$

另外，由于$x_i(i=1,\cdots,n)$只取值 0 或 1，所以还有

$$x_i(1-x_i)=0, i=1,\cdots,n.$$

最佳投资方案应是投资额最小而总收益最大的方案，所以这个最佳投资决策问题归结为总资金以及决策变量（取 0 或 1）的限制条件下，极大化总收益和总投资之比。因此，其数学模型为

$$\max Q = \frac{\sum_{i=1}^{n} b_i x_i}{\sum_{i=1}^{n} a_i x_i},$$

$$\text{s.t.} \quad \begin{cases} 0 < \sum_{i=1}^{n} a_i x_i \le A, \\ x_i(1 - x_i) = 0, i = 1, \cdots, n. \end{cases}$$

在一组等式或不等式的约束下，求一个函数的最大值（或最小值）问题，其中至少有一个非线性函数，这类问题称之为非线性规划问题。可概括为一般形式

$$\min\ f(x),$$

$$\text{s.t.}\quad \begin{cases} h_j(x) \le 0, & j = 1,\cdots,q, \\ g_i(x) = 0, & i = 1,\cdots,p. \end{cases} \tag{3.1}$$

其中$x = [x_1,\cdots,x_n]^T$称为模型（3.1）的决策变量，f称为目标函数，$g_i(i = 1,\cdots,p)$和$h_j(j = 1,\cdots,q)$称为约束函数。另外，$g_i(x) = 0\ \ (i = 1,\cdots,p)$称为等式约束，$h_j(x) \le 0$ $(j = 1,\cdots,q)$称为不等式约束。

对于一个实际问题，在把它归结成非线性规划问题时，一般要注意如下几点

（1）确定供选方案：首先要收集同问题有关的资料和数据，在全面熟悉问题的基础上，确认什么是问题的可供选择的方案，并用一组变量来表示它们。

（2）提出追求目标：经过资料分析，根据实际需要和可能，提出要追求极小化或极大化的目标。并且，运用各种科学和技术原理，把它表示成数学关系式。

（3）给出价值标准：在提出要追求的目标之后，要确立所考虑目标的“好”或“坏”的价值标准，并用某种数量形式来描述它。

（4）寻求限制条件：由于所追求的目标一般都要在一定的条件下取得极小化或极大化效果，因此还需要寻找出问题的所有限制条件，这些条件通常用变量之间的一些不等式或等式来表示。

3.1.2 非线性规划的Matlab解法

Matlab 中非线性规划的数学模型写成以下形式

$$\min \ f(x),$$

$$\text{s.t.} \quad \begin{cases} A\cdot x \le b, \\ Aeq\cdot x = beq, \\ c(x) \le 0, \\ ceq(x) = 0, \\ lb \le x \le ub. \end{cases}$$

其中 $f(x)$是标量函数，A,b,Aeq,beq,lb,ub是相应维数的矩阵和向量，$c(x),ceq(x)$是非线性向量函数。

Matlab 中的命令是

[x,fval]=fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)

x 的返回值是决策向量x的取值，fval 返回的是目标函数的取值，其中 fun 是用 M 文件定义的函数$f(x)$；x0 是x的初始值；A,b,Aeq,beq 定义了线性约束$Ax \le b, Aeq \cdot x = beq$，如果没有线性约束，则 A=[],b=[],Aeq=[],beq=[]；lb 和 ub 是变量x的下界和上界，如果上界和下界没有约束，即x无下界也无上界，则 lb=[]，ub=[]，也可以写成 lb 的各分量都为-inf，ub 的各分量都为 inf；nonlcon 是用 M 文件定义的非线性向量函数$c(x),ceq(x)$ options 定义了优化参数，可以使用 Matlab 缺省的参数设置

例 3.2　求下列非线性规划

$$
\begin{aligned}
\min\ & f(x) = x_1^2 + x_2^2 + x_3^2 + 8, \\
\text{s.t.}\ & x_1^2 - x_2 + x_3^2 \geq 0, \\
& x_1 + x_2^2 + x_3^3 \leq 20, \\
& -x_1 - x_2^2 + 2 = 0, \\
& x_2 + 2x_3^2 = 3, \\
& x_1, x_2, x_3 \geq 0.
\end{aligned}
$$

解　（1）编写 M 函数 fun1.m 定义目标函数

```
function f=fun1(x);
f=sum(x.^2)+8;
```

（2）编写M函数fun2.m定义非线性约束条件

```
function [g,h]=fun2(x);
g=[-x(1)^2+x(2)-x(3)^2
 x(1)+x(2)^2+x(3)^3-20];   %非线性不等式约束
h=[-x(1)-x(2)^2+2
 x(2)+2*x(3)^2-3]; %非线性等式约束
```

（3）编写主程序文件如下

```
[x,y]=fmincon('fun1',rand(3,1),[],[],[],[],zeros(3,1),[],'fun2')
```

求得当$x_1 = 0.5522, x_2 = 1.2033, x_3 = 0.9478$时，最小值$y = 10.6511$。

3.2　无约束问题的 Matlab 解法

3.2.1　无约束极值问题的符号解

例 3.3　求多元函数

$$f(x,y)=x^3-y^3+3x^2+3y^2-9x$$

的极值。

解　先解方程组

$$\begin{cases} f_x(x,y)=3x^2+6x-9=0 \\ f_y(x,y)=-3y^2+6y=0 \end{cases}$$

求得驻点为(1,0)，(1,2)，(−3,0)，(−3,2)。

再求出 Hessian 阵

$$\begin{bmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial x \partial y} & \dfrac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 6+6x & 0 \\ 0 & 6-6y \end{bmatrix},$$

如果在驻点处 Hessian 阵为正定阵，则在该点取极小值；如果在驻点处 Hessian 阵为负定阵，则在该点取极大值；如果在驻点处 Hessian 阵为不定阵，则该驻点不是极值点。

可以验证

点$(1,0)$是极小值点，对应的极小值 $f(1,0)=-5$；

点$(1,2)$，$(-3,0)$不是极值点；

点$(-3,2)$是极大值点，对应的极大值 $f(-3,2)=31$。

计算的 Matlab 程序如下

```
clc, clear
syms x y
f=x^3-y^3+3*x^2+3*y^2-9*x;
df=jacobian(f);   %求一阶偏导数
d2f=jacobian(df); %求 Hessian 阵
[xx,yy]=solve(df)   %求驻点
xx=double(xx);yy=double(yy);
for i=1:length(xx)
    a=subs(d2f,{x,y},{xx(i),yy(i)});
    b=eig(a);   %求矩阵的特征值
    f=subs(f,{x,y},{xx(i),yy(i)});f=double(f);
    if all(b>0)
        fprintf('(%f,%f)是极小值点,对应的极小值为%f\n',xx(i),yy(i),f);
    elseif all(b<0)
        fprintf('(%f,%f)是极大值点,对应的极大值为%f\n',xx(i),yy(i),f);
    elseif any(b>0) & any(b<0)
        fprintf('(%f,%f)不是极值点\n',xx(i),yy(i));
    else
        fprintf('无法判断(%f,%f)是否是极值点\n',xx(i),yy(i));
    end
end
```

3.2.2 无约束极值问题的数值解

在 Matlab 工具箱中，用于求解无约束极小值问题的函数有 fminunc 和 fminsearch，用法介绍如下。

求函数的极小值

$$\min_{x} f(x) ,$$

其中 x 可以为标量或向量。

Matlab 中 fminunc 的基本命令是

[x,fval]=fminunc(fun,x0,options)

其中返回值 x 是所求得的极小值点，返回值 fval 是函数的极小值。fun 是一个 M 函数，当 fun 只有一个返回值时，它的返回值是函数 $f(x)$；当 fun 有两个返回值时，它的第二个返回值是 $f(x)$的梯度向量；当 fun 有三个返回值时，它的第三个返回值是 $f(x)$的二阶导数阵（Hessian 阵）。x0 是向量 x 的初始值，options 是优化参数，可以使用缺省参数。

求多元函数的极小值也可以使用 Matlab 的 fminsearch 命令，其使用格式为

[x,fval]＝fminsearch(fun,x0,options)

例 3.4 求多元函数

$$f(x,y)=x^3-y^3+3x^2+3y^2-9x$$

的极值。

解 编写 Matlab 程序如下

```
clc, clear
f=@(x) x(1)^3-x(2)^3+3*x(1)^2+3*x(2)^2-9*x(1);
定义匿名函数
g=@(x) -f(x);
[xy1,z1]=fminunc(f, rand(2,1))    %求极小值点
[xy2,z2]=fminsearch(g,rand(2,1)); %求极大值点
xy2, z2=-z2
```

求得的极小值点为(1,0)，极小值为-5；极大值点为$(-3,2)$，极大值为 31。

例 3.5 求函数 $f(x)=100(x_2-x_1{}^2)^2+(1-x_1)^2$ 的极小值。

解　在求极小值时，可以使用函数的梯度，编写 M 函数 fun3.m 如下

```
function [f,g]=fun3(x);
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
g=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
%g 返回的是梯度向量
```

编写主程序文件如下

```
options = optimset('GradObj','on');
[x,y]=fminunc('fun3',rand(1,2),options)
```

即可求得函数的极小点 (1,1)，函数的极小值为 13.9917×10^{-15}，即极小值近似为0。

在求极值时，也可以利用二阶导数，编写 M 函数 fun4.m 如下

```
function [f,df,d2f]=fun4(x);
f=100*(x(2)-x(1)^2)^2+(1-x(1))^2;
df=[-400*x(1)*(x(2)-x(1)^2)-2*(1-x(1));200*(x(2)-x(1)^2)];
d2f=[-400*x(2)+1200*x(1)^2+2,-400*x(1)
     -400*x(1),200];
```

编写主程序文件如下

```
options = optimset('GradObj','on','Hessian','on');
[x,y]=fminunc('fun4',rand(1,2),options)
```

即可求得函数的极小值。

例 3.6 求函数 $f(x)=\sin(x)+3$ 取极小值时的 x 值

解 编写 $f(x)$ 的 M 函数 fun5.m 如下

```
function f=fun5(x);
f=sin(x)+3;
```

编写主程序文件如下

```
x0=2;
[x,y]=fminsearch(@fun5,x0)
```

求得在初值 2 附近的极小值点 $x=4.7124$ 及极小值 $y=2$

3.2.3 求函数的零点和方程组的解

例 3.7 求多项式 $f(x)=x^3-x^2+2x-3$ 的零点。

解 Matlab 程序如下

```
clc, clear
xishu=[1 -1 2 -3]; %多项式是用向量定义的，系数从高次幂到低次幂排列
x0=roots(xishu)
```

求得多项式的全部零点 $-0.1378\pm1.5273i$，1.2757。

使用符号求解的程序如下

```
syms x
x0=solve(x^3-x^2+2*x-3) %求函数零点的符号解
x0=vpa(x0,5)   %化成小数格式的数据
```

也求得全部的零点$-0.1378\pm1.5273i$，1.2757。

求数值解的 Matlab 程序如下

```
y=@(x) x^3-x^2+2*x-3;
x=fsolve(y,rand)   %只能求给定初始值附近的一个零点
```

求得给定初始值附件的一个零点1.2757。

例 3.8 求如下方程组的解

$$\begin{cases} x^2 + y - 6 = 0, \\ y^2 + x - 6 = 0. \end{cases}$$

解 符号求解的 Matlab 程序如下

```
syms x y
[x,y]=solve(x^2+y-6,y^2+x-6)
```

求得方程组的 4 组解 $(2,2)$，$(-3,-3)$，

$$\left(\frac{1+\sqrt{21}}{2},\frac{1-\sqrt{21}}{2}\right),$$

$$\left(\frac{1-\sqrt{21}}{2},\frac{1+\sqrt{21}}{2}\right)。$$

求数值解的程序如下

```
f=@(x) [x(1)^2+x(1)-6; x(2)^2+x(1)-6];
xy=fsolve(f,rand(2,1))  %只能求给定初始值附近的一组解
```

求得一组数值解(2, 2)。

3.3　约束极值问题

带有约束条件的极值问题称为约束极值问题，也叫规划问题。求解约束极值问题要比求解无约束极值问题困难得多。为了简化其优化工作，可采用以下方法：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以及能将复杂问题变换为较简单问题的其它方法。

库恩—塔克条件是非线性规划领域中最重要的理论成果之一，是确定某点为最优点的必要条件，但一般说它并不是充分条件（对于凸规划，它既是最优点存在的必要条件，同时也是充分条件）。

3.3.1 二次规划

若某非线性规划的目标函数为自变量 x 的二次函数约束条件又全是线性的，就称这种规划为二次规划。

Matlab 中二次规划的数学模型可表述如下

$$\min \ \frac{1}{2} x^T H x + f^T x,$$

$$\text{s.t.} \quad \begin{cases} Ax \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub. \end{cases}$$

这里 H 是实对称矩阵，f, b, beq, lb, ub 是列向量，A, Aeq 是相应维数的矩阵。

Matlab 中求解二次规划的命令是

[x,fval] = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0,options)

返回值 x 是决策向量 x 的值，返回值 fval 是目标函数在 x 处的值（具体细节可以参看在 Matlab 命令窗口中运行 help quadprog 后的帮助）。

例 3.9　求解二次规划

$$\min f(x) = 2x_1^2 - 4x_1x_2 + 4x_2^2 - 6x_1 - 3x_2,$$

$$\text{s.t.} \begin{cases} x_1 + x_2 \le 3, \\ 4x_1 + x_2 \le 9, \\ x_1, x_2 \ge 0. \end{cases}$$

解　编写如下程序

```
h=[4,-4;-4,8];
f=[-6;-3];
a=[1,1;4,1];
b=[3;9];
[x,value]=quadprog(h,f,a,b,[],[],zeros(2,1))
```

求得 $x_1 = 1.9500$，$x_2 = 1.0500$，$\min f(x) = -11.0250$。

3.3.2　罚函数法

利用罚函数法，可将非线性规划问题的求解，转化为求解一系列无约束极值问题，因而也称这种方法为序列无约束最小化技术，简记为 SUMT (Sequential Unconstrained Minization Technique)。

罚函数法求解非线性规划问题的思想是，利用问题中的约束函数作出适当的罚函数，由此构造出带参数的增广目标函数，把问题转化为无约束非线性规划问题。主要有两种形式,一种叫外罚函数法,另一种叫内罚函数法,下面介绍外罚函数法。

考虑问题

$$\min f(x),$$

$$\text{s.t.} \begin{cases} g_i(x) \le 0, \quad i = 1, \cdots, r, \\ h_j(x) \ge 0, \ j = 1, \cdots, s, \\ k_m(x) = 0, m = 1, \cdots, t. \end{cases}$$

一个充分大的数 $M>0$ ，构造函数

$$P(x,M)=f(x)+M\sum_{i=1}^{r}\max(g_i(x),0)$$

$$-M\sum_{i=1}^{s}\min(h_i(x),0)+M\sum_{i=1}^{t}|k_i(x)|,$$

或

$$P(x,M)=f(x)+M\,\mathrm{sum}\left(\max\begin{pmatrix}G(x)\\0\end{pmatrix}\right)$$

$$-M\,\mathrm{sum}\left(\min\begin{pmatrix}H(x)\\0\end{pmatrix}\right)+M\,\|K(x)\|,$$

这里$G(x)=[g_1(x),\cdots,g_r(x)]$，$H(x)=[h_1(x),\cdots,h_s(x)]$，$K(x)=[k_1(x),\cdots,k_t(x)]$，Matlab 中可以直接利用 max、min 和 sum 函数），则以增广目标函数$P(x,M)$为目标函数的无约束极值问题$\min P(x,M)$的最优解x也是原问题的最优解。

例 3.10　求下列非线性规划

$$\min f(x) = x_1^2 + x_2^2 + 8,$$

$$\text{s.t.} \quad \begin{cases} x_1^2 - x_2 \geq 0, \\ -x_1 - x_2^2 + 2 = 0, \\ x_1, x_2 \geq 0. \end{cases}$$

解　（1）定义增广目标函数，编写 M 函数 test1.m

```
function g=test1(x);
M=50000;
f=x(1)^2+x(2)^2+8;
g=f-M*min(x(1),0)-M*min(x(2),0)-M*min(x(1)^2-x(2),0
)+M*abs(-x(1)-x(2)^2+2);
```

或者是利用Matlab的求矩阵的极小值和极大值函数编写test2.m如下

```
function g=test2(x);
M=50000;
f=x(1)^2+x(2)^2+8;
g=f-M*sum(min([x';zeros(1,2)]))-M*min(x(1)^2-x(2),0)
+M*abs(-x(1)-x(2)^2+2);
```

也可以修改增广目标函数的定义，编写test3.m如下

```
function g=test3(x);
M=50000;
f=x(1)^2+x(2)^2+8;
g=f-M*min(min(x),0)-M*min(x(1)^2-x(2),0)+M*(-x(1)
-x(2)^2+2)^2;
```

（2）求增广函数目标函数的极小值，在 Matlab 命令窗口输入

```
[x,y]=fminsearch('test3',rand(2,1))
```

即可求得问题的解。由于是非线性问题，很难求得问题的全局最优解，只能求得一个局部最优解，并且每次的运行结果都是不一样的。

注:（1）如果非线性规划问题要求实时算法，可以使用罚函数方法，但计算精度较低。

（2）如果非线性规划问题，不要求实时算法，但要求精度高，可以使用 Lingo 软件编程求解或使用 Matlab 的 fmincon 命令求解。

3.3.3 Matlab 求约束极值问题

在 Matlab 优化工具箱中，用于求解约束最优化问题的函数有：fminbnd、fmincon、quadprog、fseminf、fminimax，上面我们已经介绍了函数 fmincon 和 quadprog。

1 fminbnd 函数

求单变量非线性函数在区间上的极小值

$$\min_{x} f(x) \ ,\ \ x \in [x_1, x_2]$$

Matlab 的命令为

[x,fval] = fminbnd(fun,x1,x2,options),

它的返回值是极小点 x 和函数的极小值。这里 fun 是用 M 文件定义的函数、匿名函数或 Matlab 中的单变量数学函数。

例 3.11　求函数 $f(x)=(x-3)^2-1,\ x\in[0,5]$ 的最小值。

解　编写 M 函数 fun6.m

```
function f=fun6(x);
f=(x-3)^2-1;
```

在 Matlab 的命令窗口输入

```
[x,y]=fminbnd('fun6',0,5)
```

即可求得极小值点 $x=3$ 和极小值 $y=$ -1。

2 fseminf 函数

$$
\min f(x),
$$

$$
\text{s.t.}\quad
\begin{cases}
A\cdot x \le b,\\
Aeq\cdot x = beq,\\
lb \le x \le ub,\\
c(x) \le 0,\\
ceq(x) \le 0,\\
K_i(x,w_i) \le 0,\ 1 \le i \le n.
\end{cases}
$$

其中x,b,beq,lb,ub是向量，A,Aeq是矩阵，$c(x),ceq(x)$是向量函数；$K_i(x,w_i)$是标量函数；$w_1,\cdots,w_n$是附加的变量。

上述问题的 Matlab 命令格式为

[x,fval]=fseminf(fun,x0,ntheta,seminfcon,A,b,Aeq,beq,lb,ub)

其中 fun 用于定义目标函数 $f(x)$；x0 为 x 的初始值；ntheta 是半无穷约束 $K_i(x,w_i)$ 的个数；函数 seminfcon 用于定义非线性不等式约束 $c(x)$，非线性等式约束 $ceq(x)$ 和半无穷约束 $K_i(x,w_i)$ 的函数，函数 seminfcon 有两个输入参量 x 和 s，s 是推荐的取样步长，也许不被使用。

例 3.12 求函数

$$f(x)=(x_1-0.5)^2+(x_2-0.5)^2+(x_3-0.5)^2$$

取最小值时的x值，约束为

$$K_1(x,w_1)=\sin(w_1x_1)\cos(w_1x_2)$$

$$-\frac{1}{1000}(w_1-50)^2-\sin(w_1x_3)-x_3\le 1,$$

$$K_2(x,w_2)=\sin(w_2x_2)\cos(w_2x_1)$$

$$-\frac{1}{1000}(w_2-50)^2-\sin(w_2x_3)-x_3\le 1,$$

$1\le w_1\le 100$， $1\le w_2\le 100.$

解　（1）编写 M 函数 fun7.m 定义目标函数如下

```
function f=fun7(x,s);
f=sum((x-0.5).^2);
```

（2）编写 M 函数 fun8.m 定义约束条件如下

```
function [c,ceq,k1,k2,s]=fun8(x,s);
c=[];ceq=[];
if isnan(s(1,1))
    s=[0.2,0;0.2 0];
end
w1=1:s(1,1):100;
w2=1:s(2,1):100;
k1=sin(w1*x(1)).*cos(w1*x(2))-1/1000*(w1-50).^2-sin(w1*x(3))-x(3)-1;
k2=sin(w2*x(2)).*cos(w2*x(1))-1/1000*(w2-50).^2-sin(w2*x(3))-x(3)-1;
plot(w1,k1,'-',w2,k2,'+');
```

（3）调用函数 fseminf

编写主程序文件如下

x0 = [0.5; 0.2; 0.3]; %如果初始值取的不合适，可能就得不到可行解

```
[x,y]=fseminf(@fun7,x0,2,@fun8)
```

求得 $x_1 = 0.6675$，$x_2 = 0.3012$，$x_3 = 0.4022$，对应的最小值 $f(x) = 0.0771$。

3 fminimax 函数

求解 $\min_{x} \max_{i} F_i(x)$，

$$
\text{s.t.} \quad \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ c(x) \le 0, \\ ceq(x) = 0, \\ lb \le x \le ub. \end{cases}
$$

的 Matlab 命令为

```
[x,fval]=fminimax(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

例 3.13 求函数族$\{f_1(x), f_2(x), f_3(x), f_4(x), f_5(x)\}$取极小一极大值时的$x$值，其中

$$\begin{cases} f_1(x) = 2x_1^2 + x_2^2 - 48x_1 - 40x_2 + 304, \\ f_2(x) = -x_1^2 - 3x_2^2, \\ f_3(x) = x_1 + 3x_2 - 18, \\ f_4(x) = -x_1 - x_2, \\ f_5(x) = x_1 + x_2 - 8. \end{cases}$$

解　（1）编写 M 函数 fun9.m 定义向量函数如下

```
function f=fun9(x);
f=[2*x(1)^2+x(2)^2-48*x(1)-40*x(2)+304
    -x(1)^2-3*x(2)^2
    x(1)+3*x(2)-18
    -x(1)-x(2)
  x(1)+x(2)-8];
```

（2）调用函数 fminimax

```
[x,y]=fminimax(@fun9,rand(2,1))
```

求得 $x_1=4$，$x_2=4$，对应的 $f_1(x)=0$，$f_2(x)=-64$，$f_3(x)=-2$，$f_4(x)=-8$，$f_5(x)=0$。

利用等式

$$\max_x \min_i F_i(x) = -\min_x \max_i(-F_i(x)),$$

也可以使用命令 fminimax 求极大—极小问题。

通过设置 options 中的 MinAbsMax 属性，fminimax 可以求解如下形式的问题

$$\min_x \max_i |F_i(x)|.$$

4 利用梯度求解约束优化问题

例 3.14 已知函数

$$f(x)=e^{x_1}(4x_1^2+2x_2^2+4x_1x_2+2x_2+1),$$

求

$$\min f(x),$$

$$\text{s.t.}\begin{cases} x_1x_2-x_1-x_2\le -1.5, \\ x_1x_2\ge -10. \end{cases}$$

分析　当使用梯度求解上述问题时，效率更高并且结果更准确。

题目中目标函数的梯度为

$$\begin{bmatrix} e^{x_1}(4x_1^2+2x_2^2+4x_1x_2+8x_1+6x_2+1) \\ e^{x_1}(4x_1+4x_2+2) \end{bmatrix}.$$

解　（1）编写 M 函数 fun10.m 定义目标函数及梯度函数

```
function [f,df]=fun10(x);
f=exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);
df=[exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+8*x(1)+6*x(2)+1);exp(x(1))*(4*x(2)+4*x(1)+2)];
```

（2）编写 M 函数 fun11.m 定义约束条件及约束条件的梯度函数

```
function [c,ceq,dc,dceq]=fun11(x);
c=[x(1)*x(2)-x(1)-x(2)+1.5;-x(1)*x(2)-10];
dc=[x(2)-1,-x(2);x(1)-1,-x(1)];
ceq=[];dceq=[];
```

（3）调用函数 fmincon，编写主程序文件如下

```
options=optimset('GradObj','on','GradConstr','on');
[x,y]=fmincon(@fun10,rand(2,1),[],[],[],[],[],[],@fun11
,options)
```

求得 $x_1=-9.5474$，$x_2=1.0474$，对应的极小值为 $y=0.0236$。

3.3.4 Matlab 优化工具箱的用户图形界面解法

Matlab 优化工具箱中的 optimtool 命令提供了优化问题的用户图形界面解法。optimtool 可应用到所有优化问题的求解，计算结果可以输出到 Matlab 工作空间中。

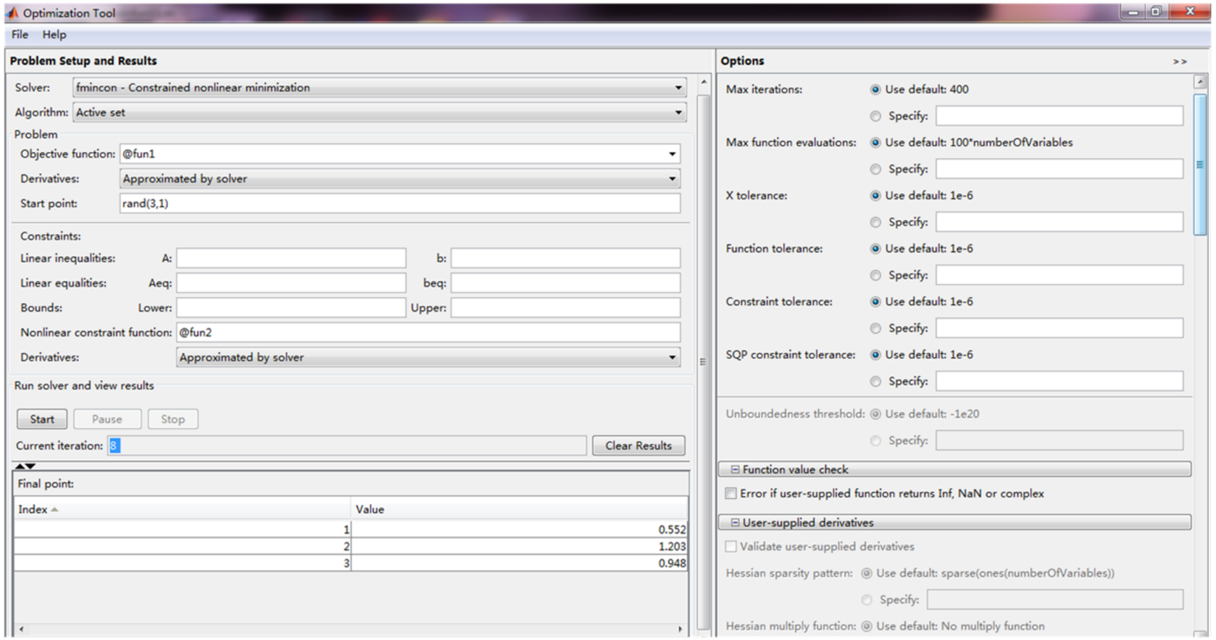

图 3.1 优化问题用户图形界面解法示意图

例 3.15　用 optimtool 重新求解例 3.2。

利用例 3.2 已经定义好的函数 fun1 和 fun2。在 Matlab 命令窗口运行 optimtool，就打开图形界面，如图 3.1 所示，填入有关的参数，未填入的参数取值为空或者为默认值，然后用鼠标点一下“start”按钮，就得到求解结果，再使用“file”菜单下的“Export to Workspace…”选项，把计算结果输出到 Matlab 工作空间中去。

3.4 飞行管理问题

在约 10，000m 高空的某边长 160km 的正方形区域内，经常有若干架飞机作水平飞行。区域内每架飞机的位置和速度向量均由计算机记录其数据，以便进行飞行管理。当一架欲进入该区域的飞机到达区域边缘时，记录其数据后，要立即计算并判断是否会与区域内的飞机发生碰撞。如果会碰撞，则应计算如何调整各架（包括新进入的）飞机飞行的方向角，以避免碰撞。现假定条件如下

（1）不碰撞的标准为任意两架飞机的距离大于 8km;

（2）飞机飞行方向角调整的幅度不应超过 30 度;

（3）所有飞机飞行速度均为每小时 800km;

（4）进入该区域的飞机在到达区域边缘时，与区域内飞机的距离应在 60km 以上;

（5）最多需考虑 6 架飞机;

（6）不必考虑飞机离开此区域后的状况。

请你对这个避免碰撞的飞行管理问题建立数学模型，列出计算步骤，对以下数据进行计算（方向角误差不超过 0.01 度），要求飞机飞行方向角调整的幅度尽量小。设该区域 4 个顶点的座标为(0,0)，(160,0)，(160,160)，(0,160)。记录数据见表 3.1。

表 3.1　飞行记录数据

飞机编号	横座标 x	纵座标 y	方向角（度）
1	150	140	243
2	85	85	236
3	150	155	220.5
4	145	50	159
5	130	150	230
新进入	0	0	52

注：方向角指飞行方向与 x 轴正向的夹角。

D为飞行管理区域的边长；

Ω为飞行管理区域，取直角坐标系使其为$[0, D] \times [0, D]$；

a为飞机飞行速度，$a = 800\text{km/h}$；

(x_i^0, y_i^0)为第i架飞机的初始位置，$i = 1, \cdots, 6$，$i = 6$对应新进入的飞机；

$(x_i(t), y_i(t))$为第i架飞机在t时刻的位置；

θ_i^0为第i架飞机的原飞行方向角，即飞行方向与x轴夹角，$0 \le \theta_i^0 < 2\pi$；

$\Delta\theta_i$为第i架飞机的方向角调整量，$-\dfrac{\pi}{6} \le \Delta\theta_i \le \dfrac{\pi}{6}$；

$\theta_i = \theta_i^0 + \Delta\theta_i$为第$i$架飞机调整后的飞行方向角。

3.4.1 模型一

根据相对运动的观点在考察两架飞机i和j的飞行时可以将飞机i视为不动而飞机j以相对速度

$$v = v_j - v_i = (a\cos\theta_j - a\cos\theta_i, a\sin\theta_j - a\sin\theta_i) \qquad (3.2)$$

相对于飞机i运动，对（3.2）式进行适当的计算可得

$$v = 2a\sin\frac{\theta_j - \theta_i}{2}(-\sin\frac{\theta_j + \theta_i}{2}, \cos\frac{\theta_j + \theta_i}{2})$$

$$= 2a\sin\frac{\theta_j - \theta_i}{2}(\cos(\frac{\pi}{2} + \frac{\theta_j + \theta_i}{2}), \sin(\frac{\pi}{2} + \frac{\theta_j + \theta_i}{2})) \quad (3.3)$$

不妨设$\theta_j \geq \theta_i$，此时相对飞行方向角为$\beta_{ij} = \frac{\pi}{2} + \frac{\theta_i + \theta_j}{2}$，见图 3.2。

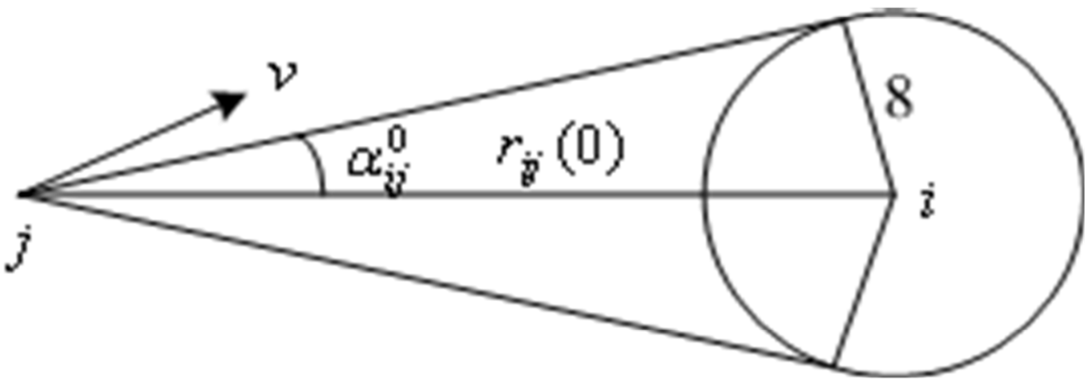

图 3.2　相对飞行方向角

由于两架飞机的初始距离为

$$r_{ij}(0)=\sqrt{(x_i^0-x_j^0)^2+(y_i^0-y_j^0)^2}\,, \tag{3.4}$$

$$\alpha_{ij}^0=\arcsin\frac{8}{r_{ij}(0)}, \tag{3.5}$$

则只要当相对飞行方向角β_{ij}**满足**

$$\alpha_{ij}^0<\beta_{ij}<2\pi-\alpha_{ij}^0 \tag{3.6}$$

时，两架飞机不可能碰撞（见图 3.2）。

记β_{ij}^0为调整前第j架飞机相对于第i架飞机的相对速度（矢量）与这两架飞机连线（从j指向i的矢量）的夹角（以连线矢量为基准，逆时针方向为正，顺时针方向为负）。则由式（3.6）知，两架飞机不碰撞的条件为

$$\left|\beta_{ij}^0+\frac{1}{2}(\Delta\theta_i+\Delta\theta_j)\right|>\alpha_{ij}^0, \tag{3.7}$$

其中

β_{mn}^{0}＝相对速度v_{mn}的幅角－从n指向m的连线矢量的幅角

$$=\arg\frac{e^{i\theta_n}-e^{i\theta_m}}{(x_m+iy_m)-(x_n+iy_n)}.$$

（注意β_{mn}^{0}表达式中的i表示虚数单位，这里为了区别虚数单位i或j，下标改写成m,n）这里我们利用复数的幅角，可以很方便地计算角度β_{mn}^{0}（$m,n=1,2,\cdots,6$）。

本问题中的优化目标函数可以有不同的形式：如使所有飞机的最大调整量最小；所有飞机的调整量绝对值之和最小等。这里以所有飞机的调整量绝对值之和最小为目标函数，可以得到如下的数学规划模型

$$\min \quad \sum_{i=1}^{6}\left|\Delta\theta_i\right|,$$

s.t.

$$\begin{cases} \left|\beta_{ij}^0+\frac{1}{2}(\Delta\theta_i+\Delta\theta_j)\right|>\alpha_{ij}^0, \quad i=1,\cdots,5, j=i+1,\cdots,6, \\ \left|\Delta\theta_i\right|\le 30^\circ, \quad i=1,2,\cdots,6. \end{cases}$$

利用如下的程序

```
clc,clear
x0=[150 85 150 145 130 0];
y0=[140 85 155 50 150 0];
q=[243 236 220.5 159 230 52];
xy0=[x0; y0];
d0=dist(xy0);     %求矩阵各个列向量之间的距离
d0(find(d0==0))=inf;
a0=asind(8./d0)   %以度为单位的反函数
xy1=x0+i*y0
xy2=exp(i*q*pi/180)
for m=1:6
    for n=1:6
        if n~=m
b0(m,n)=angle((xy2(n)-xy2(m))/(xy1(m)-xy1(n)));
        end
    end
end
b0=b0*180/pi;
dlmwrite('txt1.txt',a0,'delimiter',
'\t','newline','PC');
```

求得α_{ij}^0的值如表 3.2 所示。

表 3.2　α_{ij}^0的值

	1	2	3	4	5	6
1	0	5.39119	32.23095	5.091816	20.96336	2.234507
2	5.39119	0	4.804024	6.61346	5.807866	3.815925
3	32.23095	4.804024	0	4.364672	22.83365	2.125539
4	5.091816	6.61346	4.364672	0	4.537692	2.989819
5	20.96336	5.807866	22.83365	4.537692	0	2.309841
6	2.234507	3.815925	2.125539	2.989819	2.309841	0

求得β_{ij}^0的值如表 3.3 所示。

表 3.3 β_{ij}^{0}的值

	1	2	3	4	5	6
1	0	109.26	-128.25	24.18	173.07	14.475
2	109.26	0	-88.871	-42.244	-92.305	9
3	-128.25	-88.871	0	12.476	-58.786	0.31081
4	24.18	-42.244	12.476	0	5.9692	-3.5256
5	173.07	-92.305	-58.786	5.9692	0	1.9144
6	14.475	9	0.31081	-3.5256	1.9144	0

上述飞行管理的数学规划模型的 LINGO 程序如下

```
model:
sets:
plane/1..6/:delta;
link(plane,plane):alpha,beta;
endsets
data:
alpha=@file('txt1.txt');
beta=@file('txt1.txt');
enddata
min=@sum(plane:@abs(delta));
@for(plane:@bnd(-30,delta,30));
@for(plane(i)|i#le#5:@for(plane(j)| j#ge#i+1:
@abs(beta(i,j)+0.5*delta(i)+0.5*delta(j))>alpha(i,j)));
end
```

求得的最优解为$\Delta\theta_3 = 2.557788^\circ$，$\Delta\theta_6 = 1.0716^\circ$，其它调整角度为 0。

3.4.2 模型二

两架飞机i,j不发生碰撞的条件为

$$(x_i(t)-x_j(t))^2+(y_i(t)-y_j(t))^2>64, \tag{3.8}$$

$$1\le i\le 5,\quad i+1\le j\le 6,\quad 0\le t\le \min\{T_i,T_j\},$$

其中T_i,T_j分别表示第i,j架飞机飞出正方形区域边界的时刻。这里

$$x_i(t)=x_i^0+at\cos\theta_i,\ y_i(t)=y_i^0+at\sin\theta_i,\ i=1,2,\cdots,n$$

$$\theta_i=\theta_i^0+\Delta\theta_i,\quad |\Delta\theta_i|\le\frac{\pi}{6},\quad i=1,2,\cdots,n.$$

下面我们把约束条件（3.8）加强为对所有的时间t都成立，记

$$l_{ij}=(x_i(t)-x_j(t))^2+(y_i(t)-y_j(t))^2-64=\tilde{a}_{ij}t^2+\tilde{b}_{ij}t+\tilde{c}_{ij}$$

其中$\tilde{a}_{ij}=4a^2\sin^2\frac{\theta_i-\theta_j}{2}$，

$$\tilde{b}_{ij}=2a[(x_i(0)-x_j(0))(\cos\theta_i-\cos\theta_j)$$

$$+(y_i(0)-y_j(0))(\sin\theta_i-\sin\theta_j)]$$

$$\tilde{c}_{ij}=(x_i(0)-x_j(0))^2+(y_i(0)-y_j(0))^2-64.$$

则两架i,j飞机不碰撞的条件是

$$\Delta_{ij}=\tilde{b}_{ij}^2-4\tilde{a}_{ij}\tilde{c}_{ij}<0. \qquad (3.9)$$

这样建立如下的非线性规划模型

$$\sum_{i=1}^{6}(\Delta\theta_i)^2,$$

$$\text{s.t.}\quad \begin{cases} \Delta_{ij} < 0, \quad 1 \le i \le 5, i+1 \le j \le 6, \\ |\Delta\theta_i| \le \dfrac{\pi}{6}, \quad i = 1, 2, \cdots, 6. \end{cases}$$