第 10 章　非线性规划

司守奎

烟台市，海军航空大学

Email：sishoukui@163.com

微信：sishoukui，QQ：94339146

在数学规划模型中，若目标函数或约束条件中至少有一个为非线性的，则称这类模型为非线性规划问题。非线性规划的一般形式如下：

$$\begin{aligned}&\min\ f(\boldsymbol{x}),\\&\text{s.t.}\begin{cases}g_i(\boldsymbol{x})\le 0, & i=1,2,\cdots,m,\\ h_j(\boldsymbol{x})=0, & j=1,2,\cdots,l,\end{cases}\end{aligned}\tag{10.1}$$

其中，$\boldsymbol{x}=[x_1,x_2,\cdots,x_n]^T\in\mathbb{R}^n$ 为 n 维决策向量，$f(\boldsymbol{x})$ 为目标函数，$g_i(\boldsymbol{x})$ 和 $h_j(\boldsymbol{x})$ 为约束函数。令集合

$$S=\{\boldsymbol{x}\,|\,g_i(\boldsymbol{x})\le 0,h_j(\boldsymbol{x})=0,i=1,2,\cdots,m,j=1,2,\cdots,l\},$$

则称 S 为可行域，优化问题(10.1)可表示为 $\min\limits_{x\in S} f(\boldsymbol{x})$。特殊地，当 $S=\mathbb{R}^n$ 时，上述优化问题称为无约束优化问题。

10.1　无约束优化问题的 Matlab 求解

求解无约束优化问题常用的 Matlab 函数有 fminunc，其调用格式如下：

```
[x,fval,exitflag]=fminunc(fun,x0)
```

其中 fun 为目标函数，x0 为初始值，x 为最优解，fval 为最优值，exitflag 为算法终止标志，若 exitflag>1,则输出的结果为局部最优解；若 exitflag≤0，则输出的结果不可靠。

求解非线性无约束优化问题还可以使用 fminsearch 函数。用法与 fminunc 类似。

例 10.1　求二元函数 $f(\boldsymbol{x})=100(x_2-x_1^2)^2+(1-x_1)^2$ 的最小值，其中 $\boldsymbol{x}=[x_1,x_2]^T$。

解法一：标准算法（不使用目标函数的梯度和二阶导数阵等信息）

```
clc, clear
fx = @(x) 100*(x(2) - x(1)^2)^2 + (1-x(1))^2;  %定义目标函数
[x1,f1,flag1] = fminunc(fx, rand(1,2))
[x2,f2,flag2] = fminsearch(fx,rand(1,2))
```

解法二：非标准算法，使用目标函数的梯度信息。

目标函数 $f(\boldsymbol{x})$ 的梯度向量

$$\nabla f(\boldsymbol{x})=\begin{bmatrix}-400(x_2-x_1^2)x_1-2(1-x_1)\\200(x_2-x_1^2)\end{bmatrix}.$$

使用目标函数的梯度信息时，求得的最小点为 $x_1=1$，$x_2=1$，求得的最小值为 1.9432×10^{-11}。

```
clc, clear
options = optimoptions('fminunc','Algorithm','trust-region',...
    'SpecifyObjectiveGradient',true);
[x2,f2,flag2]=fminunc(@fun101,[1,2],options)
```

其中函数 fun101 存放在文件 fun101.m 中，函数定义如下：

```
function [f,g] = fun101(x)
f = 100*(x(2) - x(1)^2)^2 + (1-x(1))^2;  %定义目标函数
if nargout > 1                          %需要目标函数的梯度
    g = [-400*(x(2)-x(1)^2)*x(1) - 2*(1-x(1));
```

```
        200*(x(2)-x(1)^2)];
end
end
```

解法三：基于求解器的新版 Matlab 求解程序如下：

```
clc, clear
options = optimoptions('fminunc','Algorithm','trust-region',...
    'SpecifyObjectiveGradient',true);
p.options = options;
p.x0 = [-1,2];
p.objective = @fun101;
p.solver = 'fminunc';
[x,fval] = fminunc(p)
```

求得的极小点为 $x_1=1$， $x_2=1$，极小值为 1.9885×10^{-17}。

解法四：基于问题的 solve 函数求解程序如下：

```
clc, clear
p = optimproblem;        %定义目标函数最小化的优化问题
x = optimvar('x',2);     %定义问题的决策列向量
p.Objective= 100*(x(2) - x(1)^2)^2 + (1-x(1))^2;  %定义目标函数
x0.x = rand(1,2);        %决策向量的初始值取随机向量
[s, f, flag] = solve(p,x0)
xx = s.x                 %显示决策向量的值
```

求得的极小点为 $x_1=1$， $x_2=1$，极小值为 1.2326×10^{-30}。

10.2 约束优化问题的 Matlab 求解

将非线性规划(10.1)的约束按线性和非线性分开表示，则它可表示如下的标准型：

$$\min f(\boldsymbol{x}),$$
$$\text{s.t.}\begin{cases} c(\boldsymbol{x})\le 0, \\ ceq(\boldsymbol{x})=0, \\ \boldsymbol{A}_{\text{ineq}}\boldsymbol{x}\le \boldsymbol{b}_{\text{ineq}}, \\ \boldsymbol{A}_{\text{eq}}\cdot\boldsymbol{x}=\boldsymbol{b}_{\text{eq}}, \\ \boldsymbol{lb}\le\boldsymbol{x}\le\boldsymbol{ub}. \end{cases} \tag{10.2}$$

其中， $c(\boldsymbol{x})$ 和 $ceq(\boldsymbol{x})$ 为非线性向量函数。可以用 Matlab 函数 fmincon 求解(10.2)，其基本调用格式为

```
[x,fval,exitflag] = fmincon(fun,x0,Aineq,bineq,Aeq,beq,lb,ub,nonlcon)
```

其中，fun 表示目标函数 $f(\boldsymbol{x})$，x0 是预先给定的决策向量的初始值，矩阵 Aineq,列向量 bineq 对应着线性不等式约束 $\boldsymbol{A}_{\text{ineq}}\boldsymbol{x}\le\boldsymbol{b}_{\text{ineq}}$，矩阵 Aeq,列向量 beq 对应着线性不等式约束 $\boldsymbol{A}_{\text{eq}}\cdot\boldsymbol{x}=\boldsymbol{b}_{eq}$，nonlcon 表示非线性约束 $c(\boldsymbol{x}),ceq(\boldsymbol{x})$ 的函数。

例 10.2　求解优化问题

$$\max f(\boldsymbol{x})=2x_1+3x_1^2+3x_2+x_2^2+x_3,$$

$$
\text{s.t.}\begin{cases}
x_1+2x_1^2+x_2+2x_2^2+x_3\le 10,\\
x_1+x_1^2+x_2+x_2^2-x_3\le 50,\\
2x_1+x_1^2+2x_2+x_3\le 40,\\
x_1^2+x_3=2,\\
x_1+2x_2\ge 1,\\
x_1\ge 0.
\end{cases}
$$

解　（1）利用如下的 Matlab 程序：

```
%程序文件 ex10_2_1
clc, clear
A=-[1,2,0]; b=-1;
lb=[0;-inf;-inf]; x0=rand(3,1);
f=@(x)-2*x(1)-3*x(1)^2-3*x(2)-x(2)^2-x(3); %定义目标函数的匿名函数
[x,fval,flag]=fmincon(f,x0,A,b,[],[],lb,[],@fun102)
```

其中函数 fun102 存放在文件 fun101.m 中，函数定义如下：

```
function [c,ceq]=fun102(x);     %定义非线性约束函数
c=[x(1)+2*x(1)^2+x(2)+2*x(2)^2+x(3)-10
   x(1)+x(1)^2+x(2)+x(2)^2-x(3)-50
   2*x(1)+x(1)^2+2*x(2)+x(3)-40];
ceq=x(1)^2+x(3)-2;
end
```

求得最优解为

$$x_1=2.3333,\quad x_2=0.1667,\quad x_3=-3.4444,$$

目标函数的最优值 18.0833。

（2）基于求解器的 fmincon 新调用格式程序如下：

```
%程序文件 ex10_2_2
clc, clear
options = optimoptions('fmincon','Algorithm','sqp');
problem.options = options;
problem.solver = 'fmincon';
problem.objective = @(x)-2*x(1)-3*x(1)^2-3*x(2)-x(2)^2-x(3);
problem.x0 = rand(3,1);
problem.Aineq=[-1,-2,0];
problem.bineq=-1;
problem.nonlcon=@fun102;
problem.lb=[0;-inf;-inf];
[x,fval]=fmincon(problem)
```

求解结果与（1）相同。

（3）基于问题的 Matlab 求解程序如下：

```
%程序文件 ex10_2_3
clc, clear
p=optimproblem('ObjectiveSense','max');
x=optimvar('x',3,1,'LowerBound',[0;-inf;-inf])
p.Objective = 2*x(1)+3*x(1)^2+3*x(2)+x(2)^2+x(3);
x0.x = 100*rand(3,1);
p.Constraints.con1=[x(1)+2*x(1)^2+x(2)+2*x(2)^2+x(3)<=10
   x(1)+x(1)^2+x(2)+x(2)^2-x(3)<=50
```

```
    2*x(1)+x(1)^2+2*x(2)+x(3)<=40
    1-x(1)-2*x(2)<=0];
p.Constraints.con2=x(1)^2+x(3)==2;
opt=optimoptions('fmincon','Algorithm','sqp');
[s,f,flag,out]=solve(p,x0,'Options',opt)
xx=s.x
```
求解结果与（1）相同。

（4）基于问题的求解转换为基于求解器的求解。

```
clc, clear
p=optimproblem('ObjectiveSense','max');
x=optimvar('x',3,1);
p.Objective = 2*x(1)+3*x(1)^2+3*x(2)+x(2)^2+x(3);
x0.x = -100*rand(3,1);
p.Constraints.con1=[x(1)+2*x(1)^2+x(2)+2*x(2)^2+x(3)<=10
    x(1)+x(1)^2+x(2)+x(2)^2-x(3)<=50
    2*x(1)+x(1)^2+2*x(2)+x(3)<=40
    1-x(1)-2*x(2)<=0; -x(1)<=0];
p.Constraints.con2=x(1)^2+x(3)==2;

ps=prob2struct(p,"ObjectiveFunctionName","obj102",...
    "ConstraintFunctionName","con102")
ps.x0=rand(1,3);
[s,fval]=fmincon(ps)
```

例 10.3　求解非线性整数规划

$$\max\ z = x_1^2 + x_2^2 + 3x_3^2 + 4x_4^2 + 2x_5^2 - 8x_1 - 2x_2 - 3x_3 - x_4 - 2x_5,$$

$$\text{s.t.}\begin{cases} 0 \le x_i \le 99\text{且为整数},\quad i = 1,2,\cdots,5, \\ x_1 + x_2 + x_3 + x_4 + x_5 \le 400, \\ x_1 + 2x_2 + 2x_3 + x_4 + 6x_5 \le 800, \\ 2x_1 + x_2 + 6x_3 \le 200, \\ x_3 + x_4 + 5x_5 \le 200. \end{cases}$$

解　利用 Matlab 基于问题求解的 solve 函数求得的最优解为
$x_1 = 50$， $x_2 = 99$， $x_3 = 0$， $x_4 = 99$， $x_5 = 20$，最优值 $z = 51568$.

```
clc, clear
c1=[1,1,3,4,2];         %目标函数二次项系数
c2=[-8,-2,-3,-1,-2];  %目标函数一次项系数
A = [1,1,1,1,1; 1,2,2,1,6
    2,1,6,0,0; 0,0,1,1,5];    %线性不等式约束矩阵
b = [400;800;200;200];          %线性不等式约束的常数项列
x=optimvar("x",5,"LowerBound",0,"Type","integer","UpperBound",99);
prob=optimproblem("ObjectiveSense","max","Objective",c1*x.^2+c2*x);
prob.Constraints=A*x<=b;
[s,fval,flag,out]=solve(prob), xx=s.x  %显示决策变量的取值
```

本题也可以使用 Matlab 的遗传算法函数 ga 求解。

```
clc, clear
f = @(x) x(1)^2+x(2)^2+3*x(3)^2+4*x(4)^2+2*x(5)^2-8*x(1)-2*x(2)-3*x(3)-...
```

```
x(4)-2*x(5);                    %定义目标函数的匿名函数
A = [1,1,1,1,1; 1,2,2,1,6
     2,1,6,0,0; 0,0,1,1,5];     %线性不等式约束矩阵
b = [400;800;200;200];          %线性不等式约束的常数项列
[x,fv] = ga(@(x)-f(x),5,A,b,[],[],zeros(1,5),99*ones(1,5),[],[1:5])
```

10.3 非线性规划举例

例 10.4（供应与选址）建筑工地的位置（用平面坐标a,b表示，距离单位：km）及水泥日用量c（单位：t）由**表** 10.1 给出。拟建两个料场向各工地运送水泥，两个料场日储量各为 20t,问料场建在何处，使总的吨公里数最小。

表 10.1 建筑工地的位置及水泥日用量表

	1	2	3	4	5	6
a/km	1.25	8.75	0.5	3.75	3	7.25
b/km	1.25	0.75	4.75	5	6.5	7.75
c/t	3	5	4	7	6	11

解　记工地的位置为$(a_i,b_i)(i=1,2,\cdots,6)$，水泥日用量为$c_i$；拟建料场位置为$(x_j,y_j)(j=1,2)$，日储量为$e_j$，从料场$j$向工地$i$的运送量为$z_{ij}$。

建立如下的非线性规划模型：

$$\min \sum_{i=1}^{6}\sum_{j=1}^{2} z_{ij}\sqrt{(x_j-a_i)^2+(y_j-b_i)^2}\ ,$$

$$\text{s.t.}\begin{cases}\sum_{j=1}^{2} z_{ij}=c_i, & i=1,2,\cdots,6,\\ \sum_{i=1}^{6} z_{ij}\le e_j, & j=1,2,\\ z_{ij}\ge 0, & i=1,2,\cdots,6;\ j=1,2.\end{cases}$$

利用 Matlab 软件，求得拟建料场的坐标为(7.25, 7.75)，(3.2653, 5.1920)。由两个料场向 6 个工地运料方案如表 10.2 所列，总的吨千米数为 71.9352。

表 10.2 两个料场向 6 个工地运料方案

	1	2	3	4	5	6
料场 1	0	5	0	0	0	11
料场 2	3	0	4	7	6	0

```
%程序文件 ex10_4_1.m
clc, clear, d0 = load('data10_4.txt');
prob = optimproblem;
x = optimvar('x', 2, 'LowerBound', 0);
y = optimvar('y', 2, 'LowerBound', 0);
z = optimvar('z', 6, 2, 'LowerBound',0);
a = d0(1,:); b = d0(2,:); c = d0(3,:);
prob.Objective = fcn2optimexpr(@fun10_4,x,y,z,a,b);
prob.Constraints.con1 = sum(z,2)==c';
prob.Constraints.con2 = sum(z)<=20;
s0.x = 10*rand(2,1); s0.y = 10*rand(2,1);
s0.z = 10*rand(6,2);
opt=optimoptions('fmincon','Algorithm','sqp');
```

```
[sol,fval,flag,output] = solve(prob,s0,'Options',opt)
xx=sol.x, yy=sol.y, zz=sol.z  %显示决策向量的值

function obj = fun10_4(x,y,z,a,b);
obj = 0;
for i = 1:6
    for j =1:2
        obj = obj + z(i,j)*sqrt((x(j)-a(i))^2+(y(j)-b(i))^2);
    end
end
end
```

简化目标函数的完整 Matlab 程序如下：

```
%程序文件 ex10_4_2.m
clc,clear
d=load('data10_4.txt');
a=d(1,:); b=d(2,:); c=d(3,:);
p=optimproblem;
x=optimvar('x',2,'LowerBound',0);
y=optimvar('y',2,'LowerBound',0);
z=optimvar('z',6,2,'LowerBound',0);

%利用矩阵广播定义目标函数的匿名函数
f=@(x,y,z)sum(sum(z'.*sqrt((a-x).^2+(b-y).^2)));
p.Objective=fcn2optimexpr(f,x,y,z);
p.Constraints.c1=sum(z,2)==c';
p.Constraints.c2=sum(z)<=20;
s0.x=rand(2,1); s0.y=rand(2,1); s0.z=rand(6,2);
opt=optimoptions('fmincon','Algorithm','sqp');
[s,f,flag]=solve(p,s0,'Options',opt)
xx=s.x, yy=s.y, zz=s.z
```

习题 10

10.1　求二元函数 $f(\boldsymbol{x})=3x_1^2+2x_1x_2+x_2^2-4x_1+5x_2$ 的最小值。

10.2　使用 Matlab 求解下列优化问题。

$$\min e^{x_1}(4x_1^2+2x_2^2+4x_1x_2+2x_2+1),$$

$$\text{s.t.}\begin{cases} x_1+x_2=0, \\ x_1x_2-x_1-x_2+1.5\le 0, \\ -x_1x_2-10\le 0. \end{cases}$$

10.3　某炼油厂将 4 种不同含硫量的液体原料（分别记为甲、乙、丙、丁）混合生产两种产品（分别记为 A，B）。按照生产工艺的要求，原料甲、乙、丁必须首先倒入混合池中混合，混合后的液体再分别与原料丙混合生产 A，B，且要求每种产品中甲、乙、丙、丁每种原料的含量不能低于 10%。已知原料甲、乙、丙、丁的含硫量（单位：%）分别为 3，1，2，1，进货价格（单位：千元）6，8，7，5；产品 A，B 的含硫量分别不超过 2.5，1.5，售价（单位：千元）分别为 9，15。根据市场信息，原料甲、乙的供应没有限制，原料丙、丁的供应量最多为 150t、100t，产品 A，B 的市场最大需求量分别为 300t、500t，问应该怎样安排生产？