Matlab 第六章　代数方程与最优化问题的计算机求解

袁磊祺　刘志如　宋庭鉴　周子铭　岐亦铭　董溟翔　撒普尔

2020 年 12 月 24 日

目录

代数方程的求解

一元方程的图解法

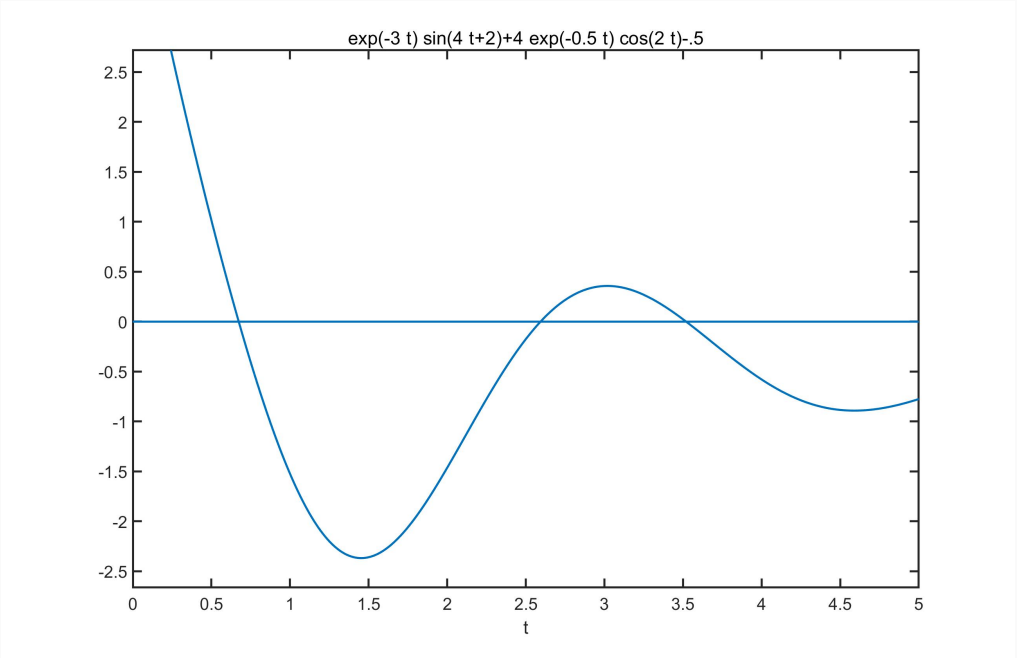
思路： 用 ezplot() 函数绘隐函数 $f(x)=0$ 曲线找零点。

例 (6-1)
解 $\mathrm{e}^{-3t}\sin(4t+2)+4\mathrm{e}^{-0.5t}\cos(2t)=0.5$。

MATLAB 代码：

```
figure
ezplot('exp(−3∗t)∗sin(4∗t+2)+4∗exp
        (−0.5∗t)∗cos(2∗t)−.5',[0 5])
line([0,5],[0,0])
```

输出：



三个解，约为 0.670、2.59、3.51.

二元方程的图解法

思路： 绘制在同一张图上看交点。

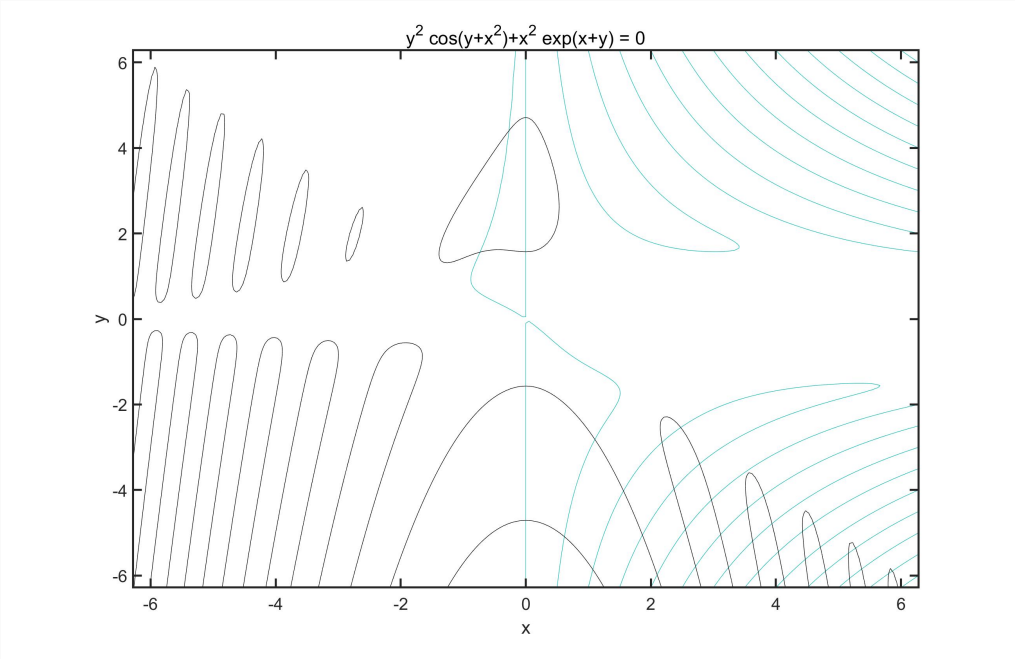
例 (6-2)

解 $\begin{cases} x^2 e^{-xy^2/2} + e^{-x/2}\sin(xy) = 0 \\ x^2\cos\left(x + y^2\right) + y^2 e^{x+y} = 0 \end{cases}$

MATLAB 代码：

```
figure
ezplot('x^2*exp(-x*y^2/2)+exp(-x/2)*sin
        (x*y)')
hold on
h1 = ezplot('y^2*cos(y+x^2)+x^2*exp(x+
        y)');
set(h1,'Color','k')
hold off
```

输出：



局限： 仅适用**一元、二元**方程。仅能**近似**得**实数**根。

多项式型方程的准解析解法

用 solve 函数，语法：

```
S = solve(eqn1, eqn2,... ,eqnn)%最简调用方式
[x,...] = solve(eqn1, eqn2,... ,eqnn)%直接得出根
[x,...] = solve(eqn1, eqn2,... ,eqnn, 'x,...')%同上, 并指定变量
```

例 (6-4)

解 $\begin{cases} x^2+y^2-1=0 \\ 0.75x^3-y+0.9=0 \end{cases}$

MATLAB 代码:

```
syms x y
[x,y]=solve(x^2+y^2−1==0,0.75∗x^3−y+0.9==0)
x=double(x), y=double(y)
[eval('x.^2+y.^2−1') eval('0.75∗x.^3−y+0.9')]%解的验算
```

输出：

```
x = −0.9817 + 0.0000i
     0.3570 + 0.0000i
    −0.5540 − 0.3547i
    −0.5540 + 0.3547i
     0.8663 − 1.2154i
     0.8663 + 1.2154i
y = 0.1904 + 0.0000i
     0.9341 + 0.0000i
     0.9293 − 0.2114i
     0.9293 + 0.2114i
    −1.4916 − 0.7059i
    −1.4916 + 0.7059i
```

一般非线性方程数值解

用 fsolve() 函数，语法：

```
x = fsolve(fun,x0)%最简单求解语句
[x,f,flag,out] = fsolve(fun,x0,opt,p1,p2,...)%一般求解格式
opt = optimset%获得默认的常用变量
opt.TolX=1e−10; set(opt,'TolX',1e−10)%修改参数
```

例 (6-8)

解 Lambert 函数，其解 w 满足 $we^w = x$。

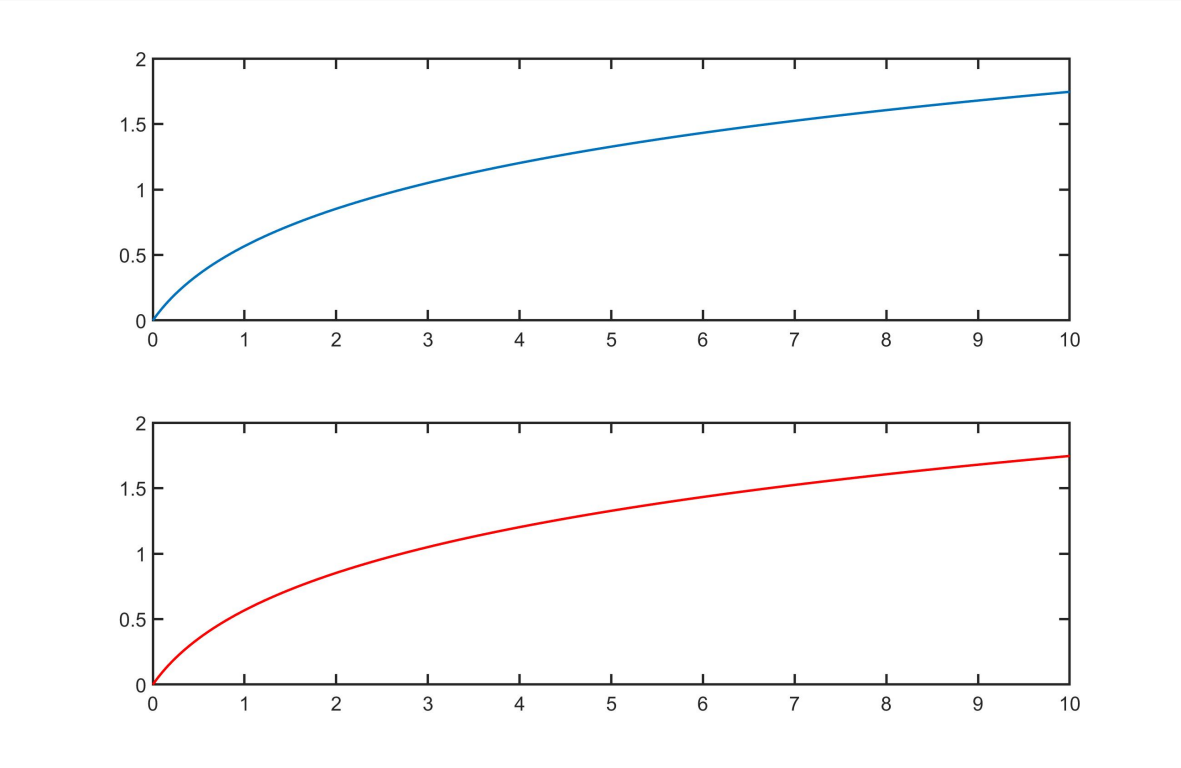
MATLAB 代码:

```
figure
y=[]; xx=0:.05:10; x0=0; h=optimset; h.Display='off';
for x = xx
y1 = fsolve(@(w) w.∗exp(w)−x,x0,h); x0=y1; y=[y,
        y1];
end
```

```
subplot(2,1,1);
plot(xx,y);
y0 = lambertw(xx);
subplot(2,1,2);
plot(xx,y0,'r');
```

一般非线性方程数值解

输出：



蓝线：fsolve() 的解，红线：软件自带该问题求值函数。

无约束最优化问题求解

解析解法与图解法

无约束最优化问题提法： $\min_x f(\mathbf{x})$, 优化变量 $\mathbf{x} = [x_1, ..., x_n]^T$，目标函数 $f(\cdot)$。

解析解法： 最优点出现在一阶导数为零的点上，可列

$$\left.\frac{\partial f}{\partial x_1}\right|_{\mathbf{x}=\mathbf{x}^*} = 0,\ \left.\frac{\partial f}{\partial x_2}\right|_{\mathbf{x}=\mathbf{x}^*} 0, \cdots,\ \left.\frac{\partial f}{\partial x_n}\right|_{\mathbf{x}=\mathbf{x}^*} = 0$$

$$\Longrightarrow \begin{cases} \text{高数：Hessian 矩阵。} \\ \textbf{图解法：} \text{画曲线判断极值点。} \checkmark \end{cases}$$

局限： 一元、二元函数适用，三元或多元函数无法画图表示。

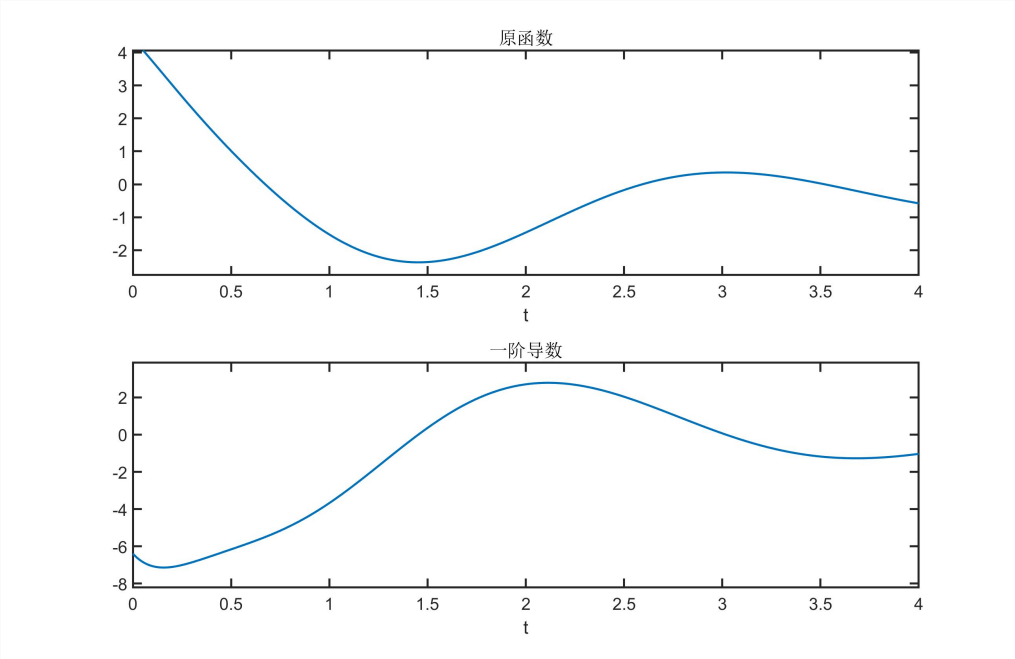
解析解法与图解法

例 (6-11)
讨论例 6-1 函数最优性。

MATLAB 代码：

```
syms t
y = exp(−3∗t)∗sin(4∗t+2)+4∗exp(−0.5∗t)∗
        cos(2∗t)−.5;
y1 = diff(y,t)
t0 = solve(y1)
b = subs(diff(y1),t,t0)
```

输出：



可见在 [0,4] 内约 1.46 处，最小值约-2.37。

数学原理： 单纯形 (Simplex) 法。(Def: N dim., N+1 vertices interconnect)

思路： 调用 fminsearch() 或者 fminunc() 函数。

调用语法，fminunc() 为例：

```
x=fminunc(Fun, x0) %最简求解语句
[x,f,flag,out]=fminunc(Fun, x0, opt, p1, p2,...) %一般求解格式
```

例 (6-12)

求 $z = f(x,y) = \left(x^2 - 2x\right) \mathrm{e}^{-x^2-y^2-xy}$ 最小值。

MATLAB 代码：

```
ff2=@(x)(x(1)^2−2∗x(1))∗exp(−x(1)^2−x(2)^2−x(1)∗x(2));
ff2_0 = optimset;
ff2_0.Display=’iter’;
result1 = fminsearch(ff2,[0; 0],ff2_0)
result2 = fminunc(ff2,[0; 0],ff2_0)
```

fminsearch 输出

```
Iteration Func−count min f(x) Procedure
0 1 0
1 3 −0.000499937 initial simplex
2 4 −0.000499937 reflect
3 6 −0.00149944 expand
4 7 −0.00149944 reflect
......
71 135 −0.641424 contract inside
72 137 −0.641424 contract outside
......
优化已终止:当前的 x 满足使用 1.000000e−04 的 OPTIONS.TolX 的终止条件，F(X) 满足使用 1.000000
        e−04 的 OPTIONS.TolFun 的收敛条件
......
result1 =
0.6111
−0.3056
```

fminunc 输出

```
Iteration Func−count f(x) Step−size First−order optimality
0 3 0 2
1 6 −0.367879 0.5 0.736
......
6 24 −0.641424 1 0.000619
7 27 −0.641424 1 1.8e−06
......
Optimization completed because the size of the gradient is less than the value of the optimality tolerance.
......
result2 =
0.6110
−0.3055
```

全局最优解与局部最优解

概念： 导数为 0 的点包括局部和全局最优点。得到的最优点依赖于初始选取。⇐ 可以用遗传算法试不同初值

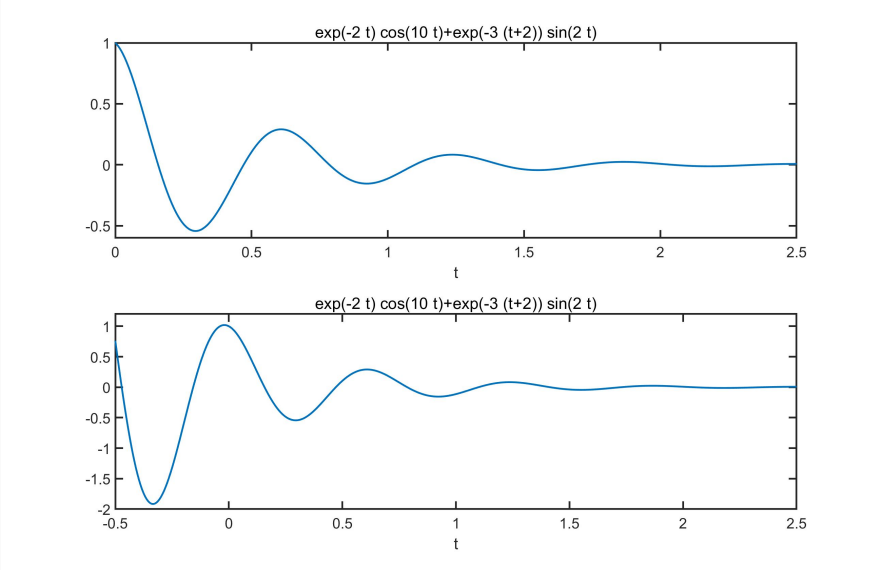
例 (6-13)

$y(t) = \mathrm{e}^{-2t}\cos 10t + \mathrm{e}^{-3t-6}\sin 2t, t \geqslant 0$ 局部最小值与全局最小值。

MATLAB 代码：

```
ff3=@(t)exp(−2∗t).∗cos(10∗t)+exp(−3∗(t
        +2)).∗sin(2∗t);
[t3_1,f3_1] = fminsearch(ff3,1)
[t3_2,f3_2] = fminsearch(ff3,.1)
```

输出：



[t3_1, f3_1] =[0.9228, -0.1547] 初值 t=1

[t3_2, f3_2] =[0.2945, -0.5436] 初值 t=.1

利用梯度求解最优化问题

思路：引入目标函数梯度，加快计算速度，改进搜索精度，但影响计算速度。即设置 optimset.GradObj=’on’ 。

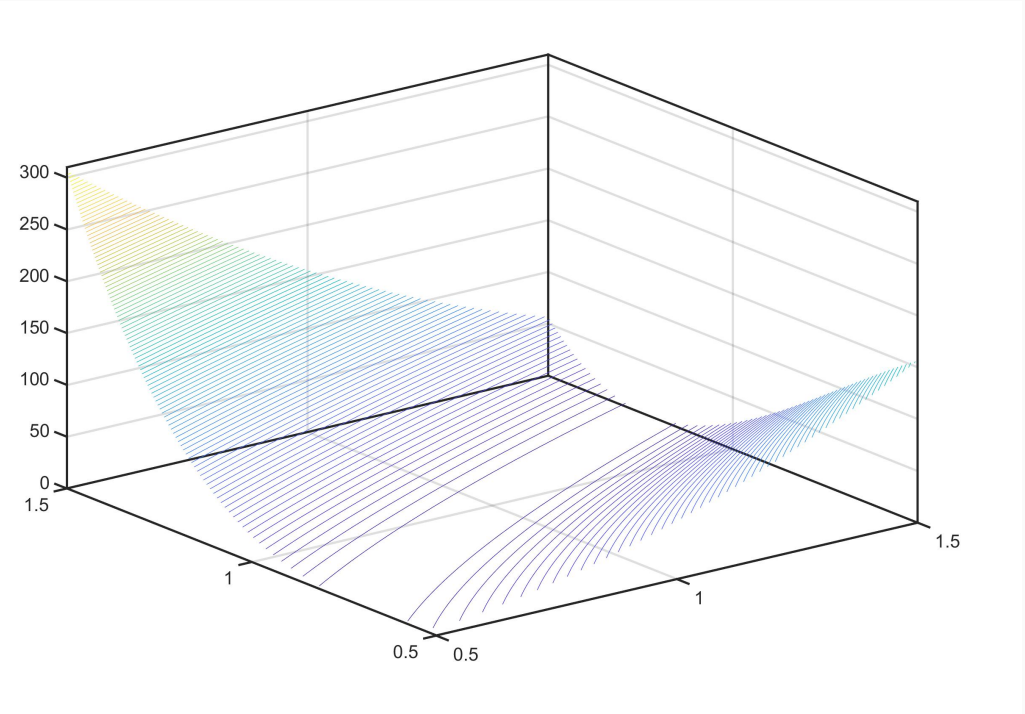
例 (6-14)

求解 Rosenbrock 函数 $f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$ 无约束最优化问题。

MATLAB 代码：

```
ff4=@(x)100*(x(2)-x(1)^2)^2+(1-x(1))^2;
ff4_0 = optimset; ff4_0.Display='iter'; ff4_0.TolX = 1e-10; ff4_0.TolFun= 1e-20;
x4 = fminunc(ff4,[0; 0],ff4_0)
syms x1 x2
ff4_1=100*(x2-x1^2)^2+(1-x1)^2;
J = jacobian(ff4_1,[x1, x2]);
```

利用梯度求解最优化问题



Rosenbrock 函数，易知最小值 (1,1)。

MATLAB 代码（续）:

```
ff4_0.GradObj='on';
x4_modi = fminunc(@ff4_modi,[0; 0],
          ff4_0)
function [y,Gy] = ff4_modi(x)
y = 100*(x(2)-x(1)^2)^2+(1-x(1))^2;
Gy = [2*x(1) - 400*x(1)*(- x(1)^2 + x(2)
          ) - 2; - 200*x(1)^2 + 200*x(2)];
end
```

利用梯度求解最优化问题

```
Iteration Func-count f(x) Step-size First-order optimality
0 3 1 2
1 12 0.771192 0.0817341 5.34
......
20 81 1.94742e-11 1 1.06e-06
x4 = (1.0000,1.0000)

Iteration Func-count f(x) Step-size First-order optimality
0 1 1 2
1 4 0.771191 0.0817342 5.34
......
22 29 1.16799e-23 1 1.26e-10
23 30 2.04561e-28 1 2.41e-13
x4_modi = (1.0000,1.0000)
```

注：实际情况常见带变量边界约束的最优化问题，如 $\min_{\mathbf{x}\ s.t.\ \mathbf{x_m}\leqslant\mathbf{x}\leqslant\mathbf{x_M}}$，可以调用 fminsearchbnd() 函数。

有约束最优化问题的计算机求解

约束条件与可行解区域

有约束最优化问题提法： $\min\limits_{\mathbf{x}\ s.t.\ \mathrm{G}(\mathbf{x})\leqslant\mathbf{0}} f(\mathbf{x})$。满足 $\mathrm{G}(\mathbf{x})\leqslant\mathbf{0}$ 的范围称为可行解区域 (feasible region)。

例 (6-15)

二元最优化问题

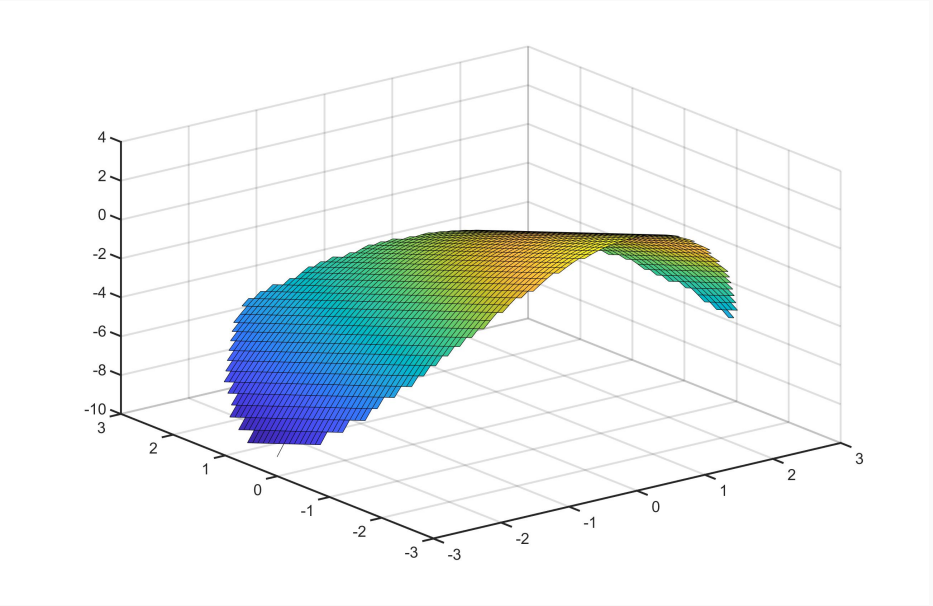
$$\max \quad \left(-x_1^2 - x_2\right)$$

$$\begin{cases} 9 \geqslant x_1^2 + x_2^2 \\ x_1 + x_2 \leqslant 1 \end{cases}$$

MATLAB 代码：

```
syms x1 x2
[x1,x2]=meshgrid(−3:.1:3);
z = −x1.^2 − x2;
i = find(x1.^2 + x2.^2 > 9); z(i) = NaN;
i = find(x1 + x2 > 1); z(i) = NaN;
surf(x1,x2,z)
```

输出：



max(z(:)) = 3, 在 (-3,0) 处。

线性规划问题的计算机求解

线性规划问题提法：

$$\min \quad \mathbf{f}^T\mathbf{x}$$
$$\mathbf{x}\ \text{s.t.} \begin{cases} \mathbf{A}\mathbf{x} \leqslant \mathbf{B} \\ \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{B}_{\text{eq}} \\ \mathbf{x}_m \leqslant \mathbf{x} \leqslant \mathbf{x}_M \end{cases},$$

线性不等式约束：$\mathbf{A}\mathbf{x} \leqslant \mathbf{B}$，线性等式约束：$\mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{B}_{\text{eq}}$，上界变量 $\mathbf{x}_M$，下界变量 $\mathbf{x}_m$。

linprog() 函数调用：

```
[x,f_opt,flag,c] = linprog(f, A, B, A_eq, B_eq, x_m, x_M, x_0, OPT, p1, p2,...)
```

线性规划问题的计算机求解

例 (6-17)

$$\min \quad (-2x_1 - x_2 - 4x_3 - 3x_4 - x_5)$$

$$\mathbf{x}\ \text{s.t.} \begin{cases} 2x_2 + x_3 + 4x_4 + 2x_5 \leqslant 54 \\ 3x_1 + 4x_2 + 5x_3 - x_4 - x_5 \leqslant 62 \\ x_1, x_2 \geqslant 0, , x_3 \geqslant 3.32, x_4 \geqslant 0.678, x_5 \geqslant 2.57 \end{cases}$$

MATLAB 代码:

```
f = [−2;−1;−4;−3;−1]; A = [0 2 1 4 2; 3 4 5 −1 −1]; B
        = [54; 62];
Ae=[]; Be=[]; xm=[0 0 3.32 .678 2.57];
ff5_0 = optimset; ff5_0.LargeScale = 'off'; ff5_0.
        Display = 'iter';
ff5_0.TolX = 1e−15; ff5_0.TolFun = 1e−10; ff5_0.
        TolCon = 1e−9;
[x, f_opt, key, c] = linprog(f, A, B, Ae, Be, xm, [], [],
        ff5_0)
```

输出:

```
Iter = 2, Time = 0.013 .....

x =
19.7850
0
3.3200
11.3850
2.5700

f_opt = −89.5750
```

二次型规划的求解

二次型规划问题提法：

$$\min \quad \left(\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}+\mathbf{f}^{\mathrm{T}}\mathbf{x}\right)$$

$$\mathbf{x}\ \text{s.t.}\ \begin{cases}\mathbf{A}\mathbf{x}\leqslant\mathbf{B}\\ \mathbf{A}_{\mathrm{eq}}\mathbf{x}=\mathbf{B}_{\mathrm{eq}}\\ \mathbf{x}_m\leqslant\mathbf{x}\leqslant\mathbf{x}_M\end{cases}$$

quadprog() 函数调用：

```
[x,f_opt,flag,c]=quadprog(H,f,A,B,A_eq,B_eq,x_m,x_M,x_0,OPT,p1,p2,...)
```

例 (6-18)

$$\min \quad \left[(x_1-1)^2+(x_2,+2)^2+(x_3-3)^2+(x_4-4)^2\right]$$

$$\mathbf{x}\ \text{s.t.}\ \begin{cases}x_1+x_2+x_3+x_4\leqslant 5\\ 3x_1+3x_2+2x_3+x_4\leq 10\\ x_1,x_2,x_3,x_4\geqslant 0\end{cases}$$

输出：

```
Iter = 5..... x =(0.0000,0.6667,1.6667,2.6667); f_opt =−23.6667
```

一般非线性规划问题的求解

一般非线性规划问题提法

$$\min \quad f(\mathbf{x})$$
$$\mathbf{x}\ \text{s.t.} \begin{cases} \mathbf{Ax} \leqslant \mathbf{B} & , \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{B}_{\text{eq}} \\ \mathbf{x}_m \leqslant \mathbf{x} \leqslant \mathbf{x}_M & , \mathbf{C}(\mathbf{x}) \leqslant 0 \\ \mathbf{C}_{eq}(\mathbf{x}) = 0 & . \end{cases}$$

fmincon() 函数调用：

```
[x,f_opt,flag,c]=fmincon(F,x0,A,B,A_eq,B_eq,x_m,x_M,CF,OPT,p1,p2,...)
%全局最优解尝试，则调fmincon_global()函数，规则同上。
```

例 (6-19)

$$\min \quad \left[1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3\right]$$
$$\mathbf{x}\ \text{s.t.} \begin{cases} x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ 8x_1 + 14x_2 + 7x_3 - 56 = 0 \\ x_1, x_2, x_3 \geqslant 0 \end{cases}$$

一般非线性规划问题的求解

MATLAB 代码

```
ff8_0 = optimset; ff8_0.LargeScale = 'off'; ff8_0.Display = 'iter';
ff8_0.TolX = 1e−15; ff8_0.TolFun = 1e−30; ff8_0.TolCon = 1e−20;
x0=ones(3,1); xm=zeros(3,1); xM=[]; A=[]; B=[]; Aeq=[]; Beq=[];
[x,f_opt,flag,c] = fmincon(@opt_fun1, x0, A, B, Aeq, Beq, xm, xM, @opt_con1, ff8_0)
```

输出

```
x =
3.5121
0.2170
3.5522

f_opt =
961.7152

iterations: 16
```

学习心得 i

一　这是本系统具体的书，可以了解，明白 MATLAB 能解决的问题，有实际需要时再详细翻阅。

二　MATLAB 集成很多科学计算算法，也方便调用。我联想到了 FFTW、LAPACK 等，让用户避免重复造轮子。但我认为需要对算法有些了解，定性明白它的长处和不足。

混合整数规划问题的计算机求解

整数规划 i

在很多应用领域中，最优化问题需要额外要求全部或部分自变量取整数，这类问题又称为整数规划.

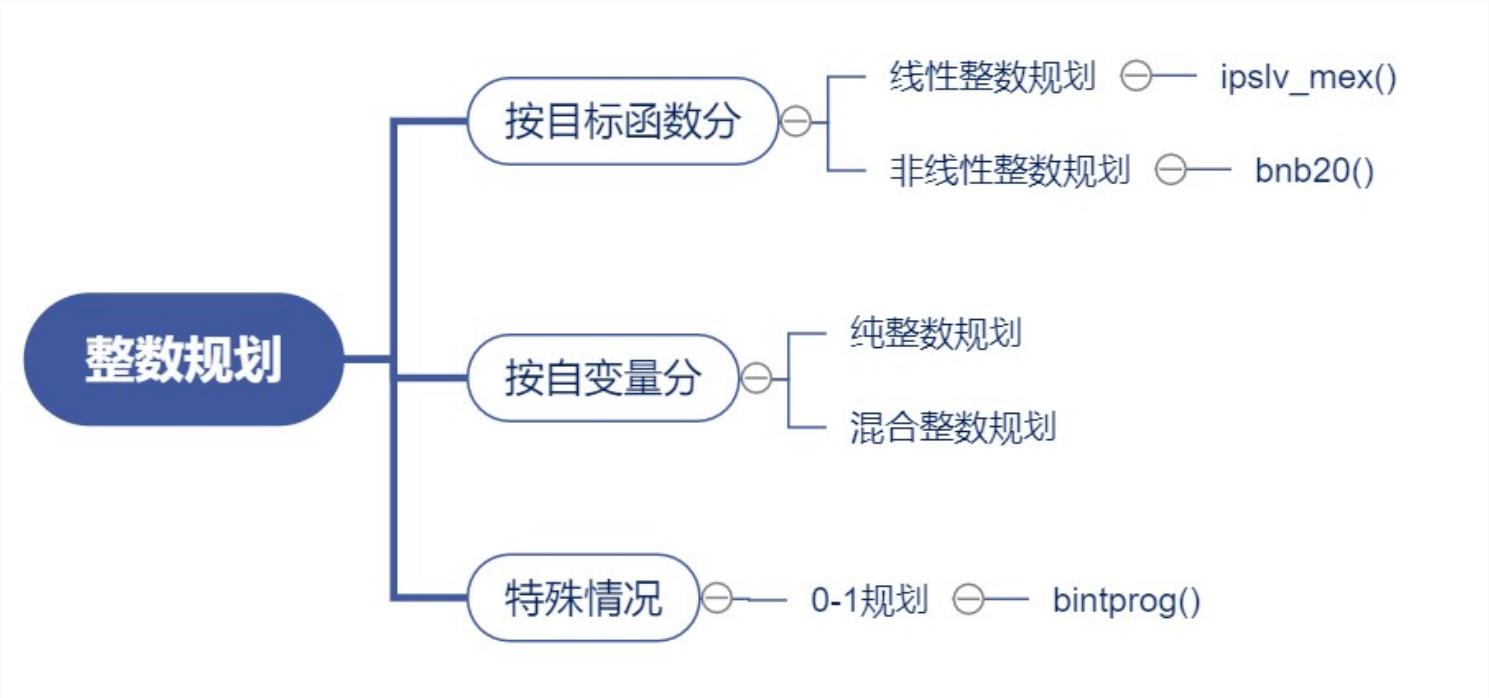


图 1: 整数规划的划分

整数线性规划问题的求解 i

$$\min f^T x \quad x \in Z \text{ s.t.} \begin{cases} Ax \leq B \\ A_{eq}x = B_{eq} \\ x_m \leq x \leq x_M \end{cases} \tag{1}$$

之前 MATLAB 没有提供整数线性规划的函数，常用有代表性的 Michel Berkelaar 等人开发的 lpslv 包.

$$[x, how] = ipslv\_\text{mex}(A, B, f, \text{intlist}, x_m, x_M, \text{ctype}) \tag{2}$$

一般非线性整数规划问题与求解 i

实际应用中经常需要求解非线性整数规划或混合规划问题，常用求解函数是基于分枝定界算法编写的 bnb20 函数.

$$[err, f, x] = bnb20\left(fun, x_0, \text{ intlist }, x_m, x_M, A, B, A_{eq}, B_{eq}\right) \tag{3}$$

该函数主要调用的是最优化工具箱函数.

0-1 规划问题求解 i

MATLAB 提供了用来求解 0-1 线性规划问题的函数 bintprog，但不能直接求解非线性 0-1 规划问题.

$$x = \text{bintprog}\left(f, A, B, A_{eq}, B_{eq}\right) \tag{4}$$

免费的 MAILAB 函数求解非线性混合整数规划的功能不是很强大，基于 MATLAB 的商业软件 TOMLAB 更复杂，但功能也更加全面且出色.

应用举例 i

MATLAB 后续版本中更新了 MILP 函数，可以求解混合线性整数规划问题.

$$x = \text{intlinprog}\left(f, \text{intcon}, A, B, A_{eq}, B_{eq}, x_m, x_M\right) \tag{5}$$

以解下面的优化问题为例

$$\begin{gathered} \min\left(-2x_1 - x_2 - 4x_3 - 3x_4 - x_5\right) \\ 2x_2 + x_3 + 4x_4 + 2x_5 \leq 54 \\ 3x_1 + 4x_2 + 5x_3 - x_4 - x_5 \leq 62 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 3.32, x_4 \geq 0.678, x_5 \geq 2.57 \end{gathered} \tag{6}$$

结果 i

$$\begin{gathered}\min\left(-2x_1 - x_2 - 4x_3 - 3x_4 - x_5\right) \\ 2x_2 + x_3 + 4x_4 + 2x_5 \leq 54 \\ 3x_1 + 4x_2 + 5x_3 - x_4 - x_5 \leq 62 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 3.32, x_4 \geq 0.678, x_5 \geq 2.57\end{gathered} \tag{7}$$

```
>> f=-[2 1 4 3 1]';A=[0 2 1 4 2;3 4 5 -1 -1];B=[54;62];
>> intcon=[1 4 5]';Aeq=[];Beq=[];
>> xm=[0 0 3.32 0.678 2.57]';xM=inf*ones(5,1);
>> x=intlinprog(f,intcon,A,B,Aeq,Beq,xm,xM);
```

图 2: 参数输入

结果 ii

```
>> x'

ans =

   19.0000         0    3.8000   11.0000    3.0000
```

图 3: 算例结果

时间客观上是连续的，但社会时间是离散的，整数规划相较于连续优化问题，看似是简化了条件，实际的解决难度却并没有降低，并且其在交通、能源等诸多实际领域中具有更高的应用价值。但是也不能否定了连续优化的作用，科学的本质便是从简到难，先把简单问题研究透彻，再把复杂问题简化为求解一个个简单的问题.

线性矩阵不等式问题求解

6.5.1 线性矩阵不等式 i

线性矩阵不等式具有一般的描述形式

$\boldsymbol{F}(x) = \boldsymbol{F}_0 + x_1\boldsymbol{F}_1 + \cdots + x_m\boldsymbol{F}_m < 0$

其中：$\boldsymbol{x} = [x_1, x_2, \cdots x_m,]^T$，为多项式向量，$\boldsymbol{F}_1$ 为实对称矩阵或者复 Hermite 矩阵。

矩阵小于零表示矩阵负定。有 $\boldsymbol{x}$ 解集为凸集，即：

$\boldsymbol{F}\left[\alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2\right] = \alpha\boldsymbol{F}(\boldsymbol{x}_1) + (1-\alpha)\boldsymbol{F}(\boldsymbol{x}_2) < 0$

其中 $0 < \alpha < 1$，该解称为可行解。

6.5.1 线性矩阵不等式 i

若有 $\boldsymbol{F}_1(x) < 0$ 与 $\boldsymbol{F}_2(x) < 0$ 则

$$\begin{bmatrix} \boldsymbol{F}_1(x) & 0 \\ 0 & \boldsymbol{F}_2(x) \end{bmatrix} < 0$$

同理有系列 $\boldsymbol{F}_i(x) < 0, i = 1, 2, ..., k$ 可有

$$\boldsymbol{F}(x) = \begin{bmatrix} \boldsymbol{F}_1(x) & & & \\ & \boldsymbol{F}_2(x) & & \\ & & \cdots & \\ & & & \boldsymbol{F}_k(x) \end{bmatrix} < 0$$

6.5.2Lyapunov 不等式 i

若给定正定矩阵 $\boldsymbol{Q}$，有 Lyapunov 方程：
$\boldsymbol{A}^T\boldsymbol{X}+\boldsymbol{X}\boldsymbol{A}=-\boldsymbol{Q}$
若存在正定的解 $\boldsymbol{X}$ 则该系统稳定。上述问题很自然地表示成对 Lyapunov 不等式的求解

$$\boldsymbol{A}^T\boldsymbol{X}+\boldsymbol{X}\boldsymbol{A}<0$$

$\boldsymbol{X}$ 是对称矩阵，可以用一个 $n\left(n+1\right)/2$ 哥元素组成的向量 $\boldsymbol{x}$ 描述：

$$x_{i-j+1+(2n-j+2)(j-1)/2}=X_{i,j}$$

其中 $j=1,..,n$ 而 $i=j,...,n$。

6.5.2Lyapunov 不等式 i

接着便可以将

$$\boldsymbol{A}^T\boldsymbol{X}+\boldsymbol{X}\boldsymbol{A}<0$$

转化为

$$\begin{bmatrix} \boldsymbol{F}_1 & & & \\ & \boldsymbol{F}_2 & & \\ & & \cdots & \\ & & & \boldsymbol{F}_{\frac{n(n+1)}{2}} \end{bmatrix} \cdot \boldsymbol{x} = \boldsymbol{F}(\boldsymbol{x}) < 0$$

matlab 中用 F

6.5.2 一般代数 Riccati 不等式 i

我们知道若

$$\boldsymbol{F}(\boldsymbol{x})=\begin{bmatrix}\boldsymbol{F}_{11}(\boldsymbol{x}) & \boldsymbol{F}_{12}(\boldsymbol{x})\\ \boldsymbol{F}_{21}(\boldsymbol{x}) & \boldsymbol{F}_{22}(\boldsymbol{x})\end{bmatrix}$$

是负定的，则有一下三个等价表达：

$$\boldsymbol{F}(\boldsymbol{x})<0$$

$$\boldsymbol{F}_{11}(\boldsymbol{x})<0, \boldsymbol{F}_{22}(\boldsymbol{x})-\boldsymbol{F}_{21}(\boldsymbol{x})\boldsymbol{F}_{11}^{-1}(\boldsymbol{x})\boldsymbol{F}_{12}(\boldsymbol{x})<0$$

$$\boldsymbol{F}_{22}(\boldsymbol{x})<0, \boldsymbol{F}_{11}(\boldsymbol{x})-\boldsymbol{F}_{12}(\boldsymbol{x})\boldsymbol{F}_{22}^{-1}(\boldsymbol{x})\boldsymbol{F}_{21}(\boldsymbol{x})<0$$

6.5.2 一般代数 Riccati 不等式 i

对于更复杂的 Riccati 方程变换可以得到 Riccati 不等式

$$\boldsymbol{A}^T\boldsymbol{X}+\boldsymbol{X}\boldsymbol{A}+\left(\boldsymbol{X}\boldsymbol{B}-\boldsymbol{C}^T\right)\boldsymbol{R}^{-1}\left(\boldsymbol{X}\boldsymbol{B}-\boldsymbol{C}^T\right)^T<0$$

对比

$$\boldsymbol{F}_{22}\left(\boldsymbol{x}\right)<0,\boldsymbol{F}_{11}\left(\boldsymbol{x}\right)-\boldsymbol{F}_{12}\left(\boldsymbol{x}\right)\boldsymbol{F}_{22}^{-1}\left(\boldsymbol{x}\right)\boldsymbol{F}_{21}\left(\boldsymbol{x}\right)<0$$

可等价构建为

$$\boldsymbol{X}>0,\begin{bmatrix}\boldsymbol{A}^T\boldsymbol{X}+\boldsymbol{X}\boldsymbol{A} & \boldsymbol{X}\boldsymbol{B}-\boldsymbol{C}^T\\ \boldsymbol{B}^T\boldsymbol{X}-\boldsymbol{C} & -\boldsymbol{R}\end{bmatrix}<0$$

6.5.3 线性矩阵不等式问题分类 i

(1) 可行性问题, 即:

$$\boldsymbol{F}(\boldsymbol{x}) < 0$$

存在解

(2) 线性目标函数最优化问题，即求:

$$\begin{aligned} &\min \boldsymbol{c}^T\boldsymbol{x} \\ \boldsymbol{x}\ &s.t. \boldsymbol{F}(\boldsymbol{x}) < 0 \end{aligned}$$

(3) 广义特征值最优化问题，即求:

$$\lambda\ \mathbf{x}\ s.t. \begin{cases} \min \lambda \\ \boldsymbol{A}(\boldsymbol{x}) < \lambda \boldsymbol{B}(\boldsymbol{x}) \\ \boldsymbol{B}(\boldsymbol{x}) > 0 \\ \boldsymbol{C}(\boldsymbol{x}) < 0 \end{cases}$$

6.5.4 线性矩阵问题 matlab 求解 i

考虑 Riccati 不等式

$$A^TX+XA+(XB)R^{-1}B^TX+Q<0$$

分块为

$$\begin{bmatrix} A^TX+XA+Q & XB \\ B^TX & -R \end{bmatrix}<0$$

```
A=[-2,-3,-1;-3,-1,-1;1,0,-4];
B=[-1,0;0,-1;-1,-1];
Q=[-2,1,-2;1,-2,-4;-2,-4,-2];
R=eye(2);

setlmis([]);
X=lmivar(1,[3 1]);
lmiterm([1 1 1 X],A',1,'s');
lmiterm([1 1 1 0],Q);
lmiterm([1 1 2 X],1,B);
lmiterm([1 2 2 0],-1);
lmiterm([-2,1,1,X],1,1);
G=getlmis;
[tmin b]=feasp(G);X=dec2mat(G,b,X);
```

图 4:

6.5.4 线性矩阵问题 matlab 求解 i

带入：$\boldsymbol{A}=\begin{bmatrix}-2 & -3 & -1\\ -3 & -1 & -1\\ 1 & 0 & -4\end{bmatrix}$

$\boldsymbol{B}=\begin{bmatrix}-1 & 0\\ 0 & -1\\ -1 & -1\end{bmatrix}$ $\boldsymbol{Q}=\begin{bmatrix}-2 & 1 & -2\\ 1 & -2 & -4\\ -2 & -4 & -2\end{bmatrix}$ $\boldsymbol{R}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$

得到

$$\boldsymbol{X}=\begin{bmatrix}0.9101 & 0.4641 & -0.2354\\ 0.4641 & 0.7893 & -0.0383\\ -0.2354 & -0.0383 & 1.3616\end{bmatrix}$$

6.5.5 基于 Yalmip 工具箱最优化求解方法 i

YALMIP 工具箱解决线性规划;

$$\min$$

$$6x_1 - 3x_2 - 5x_3 - 2x_4 - 9x_5$$

$$\mathbf{x}\ s.t. \begin{cases} 2x_2 + x_3 + 4x_4 + 2x_5 \leq 54 \\ 3x_1 + 4x_2 + 5x_3 - x_4 - x_5 \leq 66 \\ x_1 \geq 0, x_2 \geq 2, x_3 \geq 3, x_4 \geq 0.5, x_5 \geq 2 \end{cases}$$

得到

$$x_1 = 22, x_2 = 2, x_3 = 3, x_4 = 0.5, x_5 = 22.5$$

$$6x_1 - 3x_2 - 5x_3 - 2x_4 - 9x_5 = -356.5$$

```
x=sdpvar(5,1); A=[0 2 1 4 2;3 4 5 -1 -1]; B=[54;66];
xm=[0,2,3,0.5,2]';
F= [A*x <= B, xm<=x];
sol=solvesdp(F,-[6 3 5 2 9]*x); x=double(x);
Ans=-[6 3 5 2 9]*x;
```

图 5:

多目标优化模型

多目标优化模型 i

例 6-58 设商店有糖果 A1,A2,A3，价格分别为 4，2.8，2.4 元/kg。要求购买不超过 20 元，总重不超过 6kg。A1+A2 总重不少于 3kg，如何购买?
应该设立两个目标函数，一个是花钱最少，一个是重量最总，其他条件可以认为是约束条件。假设购买三种糖果的重量分别为 x1, x2, x3kg，这时目标函数分别为
花钱：$f_1(x) = 4x_1 + 2.8x_2 + 2.4x_3 \to min$
重量：$f_2(x) = x_1 + x_2 + x_3 \to \max$ 那么模型该如何设立呢?

多目标优化模型 i

$$\min \begin{bmatrix} 4x_1 + 2.8x_2 + 2.4x_3 \\ x_1 + x_2 + x_3 \end{bmatrix}$$

$$\boldsymbol{x}\ s.t. \begin{cases} 4x_1 + 2.8x_2 + 2.4x_3 \leq 20 \\ x_1 + x_2 + x_3 \geq 6 \\ x_1 + x_2 \geq 3 \\ x_1, x_2, x_3 \geq 0 \end{cases}$$

从这之中我们可以得出多目标优化模型的一般表示形式：

$$J = \min \boldsymbol{F}(\boldsymbol{x})$$

$$\boldsymbol{x}\ s.t. \boldsymbol{G}(\boldsymbol{x}) \leq 0$$

其中，$\boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x}), \cdots, f_p(\boldsymbol{x})]^T$

多目标问题转化为单目标问题求解 i

那么如何解这类问题呢?

下面介绍三种转换方法:

1. 线性加权变换及求解
2. 线性规划问题的最佳妥协解
3. 线性规划问题的最小二乘解

(1) 线性加权变换及求解 i

根据两个指标的侧重情况引入加权，目标函数改写为:

$$f(\boldsymbol{x}) = w_1 f_1(\boldsymbol{x}) + w_2 f_2(\boldsymbol{x}) + w_3 f_3(\boldsymbol{x}) + \cdots + w_p f_p(\boldsymbol{x})$$

其中，$w_1 + w_2 + \cdots + w_p = 1, 0 \leq w_1, w_2, \cdots, w_p \leq 1.$
则例 6-58 就可以改写成下面的线性规划系数

$$\begin{aligned} &\min && (w_1[4, 2.8, 2.4] - w_2[1, 1, 1])\boldsymbol{x} \\ &\boldsymbol{x}\ s.t. \begin{cases} 4x_1 + 2.8x_2 + 2.4x_3 \leq 20 \\ x_1 + x_2 + x_3 \geq 6 \\ x_1 + x_2 \geq 3 \\ x_1, x_2, x_3 \geq 0 \end{cases} \end{aligned}$$

(1) 线性加权变换及求解 ii

$$
C = \begin{bmatrix}
0 & 1.0000 & 0 & 3.00 & 4.8333 & 20.00 & 7.8333 \\
0.1000 & 0.9000 & 0 & 3.00 & 4.8333 & 20.00 & 7.8333 \\
0.2000 & 0.8000 & 0 & 3.00 & 4.8333 & 20.00 & 7.8333 \\
0.3000 & 0.7000 & 0 & 3.00 & 3.00 & 15.60 & 6.0000 \\
0.4000 & 0.6000 & 0 & 3.00 & 3.00 & 15.60 & 6.0000 \\
0.5000 & 0.5000 & 0 & 3.00 & 3.00 & 15.60 & 6.0000 \\
0.6000 & 0.4000 & 0 & 3.00 & 3.00 & 15.60 & 6.0000 \\
0.7000 & 0.3000 & 0 & 3.00 & 3.00 & 15.60 & 6.0000 \\
0.8000 & 0.2000 & 0 & 3.00 & 3.00 & 15.60 & 6.0000 \\
0.9000 & 0.1000 & 0 & 3.00 & 3.00 & 15.60 & 6.0000 \\
1.0000 & 0 & 0 & 3.00 & 3.00 & 15.60 & 6.0000
\end{bmatrix} \tag{8}
$$

(2) 线性规划问题的最佳妥协解 i

考虑一类特殊的线性规划问题

$$\boldsymbol{J} = \max \quad \boldsymbol{Cx}$$

$$\boldsymbol{x}\ s.t. \begin{cases} Ax \leq B \\ A_{eq}x = B_{eq} \\ x_m \leq x \leq x_M \end{cases}$$

目标函数的 $\boldsymbol{C}$ 不是一个向量，而是一个矩阵。
每一个目标函数 $j_i(x) = c_i x, i = 1, 2, \cdots, p$, 可以理解为第 i 方的利益分配，所以这样的最优化问题可以认为是各方利益的最大分配。
最佳妥协解的求解步骤如下：

1. 单独求解每个单目标函数的最优化问题，得出最优解 $f_k, k = 1, 2, \cdots, p$

(2) 线性规划问题的最佳妥协解 ii

2. 通过规范化构造单独的目标函数

$$f(x) = -\frac{1}{f_1}c_1x - \frac{1}{f_2}c_2x - \cdots - \frac{1}{f_p}c_px$$

3. 最佳妥协解可以变换成下面的单目标限性规划问题并直接求解

$$\boldsymbol{J} = \quad \min \quad \boldsymbol{f(x)}$$

$$\boldsymbol{x}\ s.t. \begin{cases} Ax \leq B \\ A_{eq}x = B_{eq} \\ x_m \leq x \leq x_M \end{cases}$$

$$x = \begin{bmatrix} 0 \\ 3.0000 \\ 4.8333 \end{bmatrix}, \quad \text{ans} = \begin{bmatrix} -20.0000 \\ 7.8333 \end{bmatrix} \tag{9}$$

(3) 线性规划问题的最小二乘解 i

考虑下面多目标线性规划问题的最小二乘表示

$$\min \quad \frac{1}{2}||Cx-d||^2$$

$$\boldsymbol{x}\ s.t.\begin{cases} Ax \le B \\ A_{eq}x = B_{eq} \\ x_m \le x \le x_M \end{cases}$$

则最小二乘解可以由 $x = lsqlin(C, d, A, B, A_{eq}, B_{eq}, x_m, x_M, x_0, options)$ 函数直接得到。

$$x = \begin{bmatrix} 0.0000 \\ 3.0000 \\ 3.0000 \end{bmatrix}, \quad \text{ans} = \begin{bmatrix} -15.6000 \\ 6.0000 \end{bmatrix} \tag{10}$$

动态规划及其在路径规划中的应用

动态规划及其在路径规划中的应用 i

图的数学表示 (存储方式)

有向图最短路径问题的手工求解

graphshortestpath 函数

Dijkstra 算法

图的矩阵表示法 i

关联矩阵：存储起点，终点和权值。

命令语句：

$a = [a_1, a_2, ..., a_m, n]$; 起点

$b = [b_1, b_2, ..., b_m, n]$; 终点

$w = [w_1, w_2, ..., w_m, 0]$; 权

$R = sparse(a, b, w)$; 关联矩阵的稀疏矩阵表示

最短路径问题手工求解 i

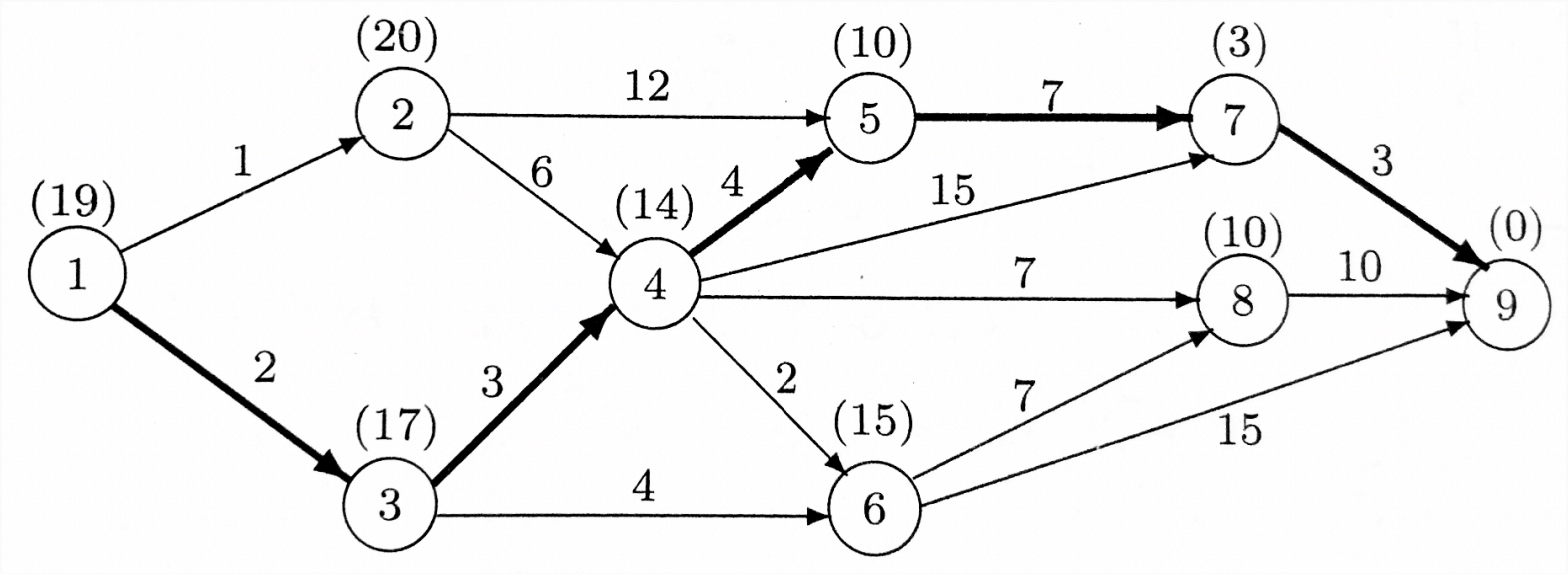


图 6: 有向图最短路径问题的手工求解

有向图搜索及图示 i

命令语句:

$P = biograph(R)$

$[d, p] = graphshortestpath(R, n_1, n_2)$, 其中 n_1 为起点， n_2 为终点。

例题解法:

$ab = [112233444456678];$

$bb = [2354465786789 99];$

$w = [12126344157277153 10];$

R=sparse(ab,bb,w); 建立关联矩阵

R(9,9)=0;

h=view(biograph(R,[],'ShowWeights','on')) 显示有向图

有向图搜索及图示 ii

[d,p]=graphshortestpath(R,1,9) 计算最短路径，返回最短路径长度和路径节点

set(h.Nodes(p),’Color’,[1 0 0]) 将路径节点标记成红色

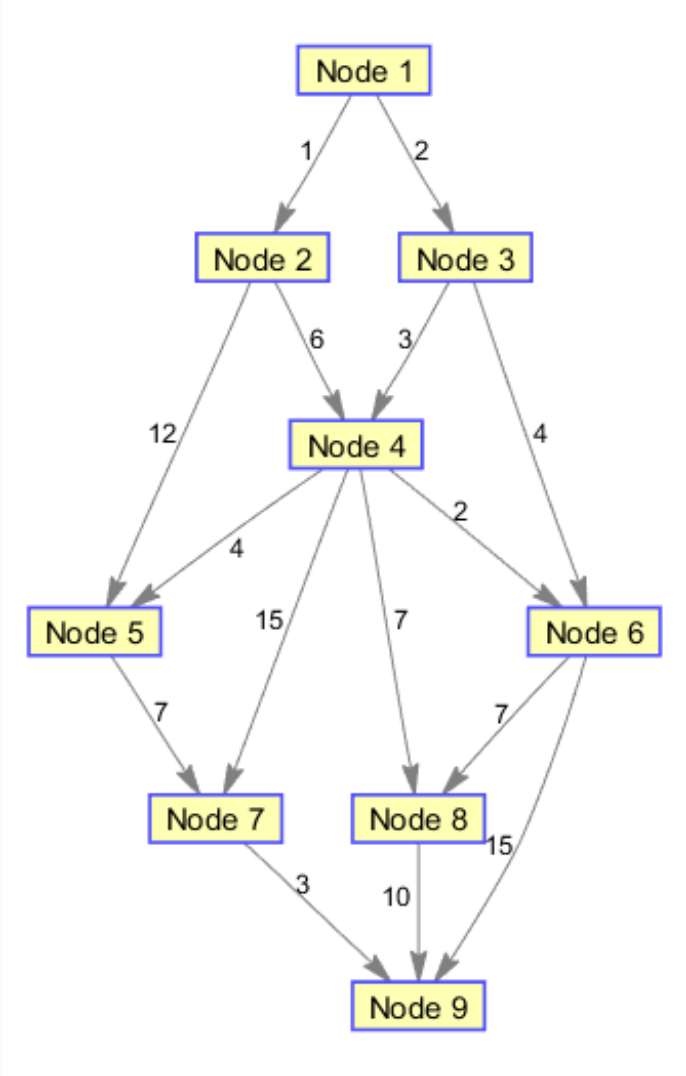


图 7: 有向图的自动绘制

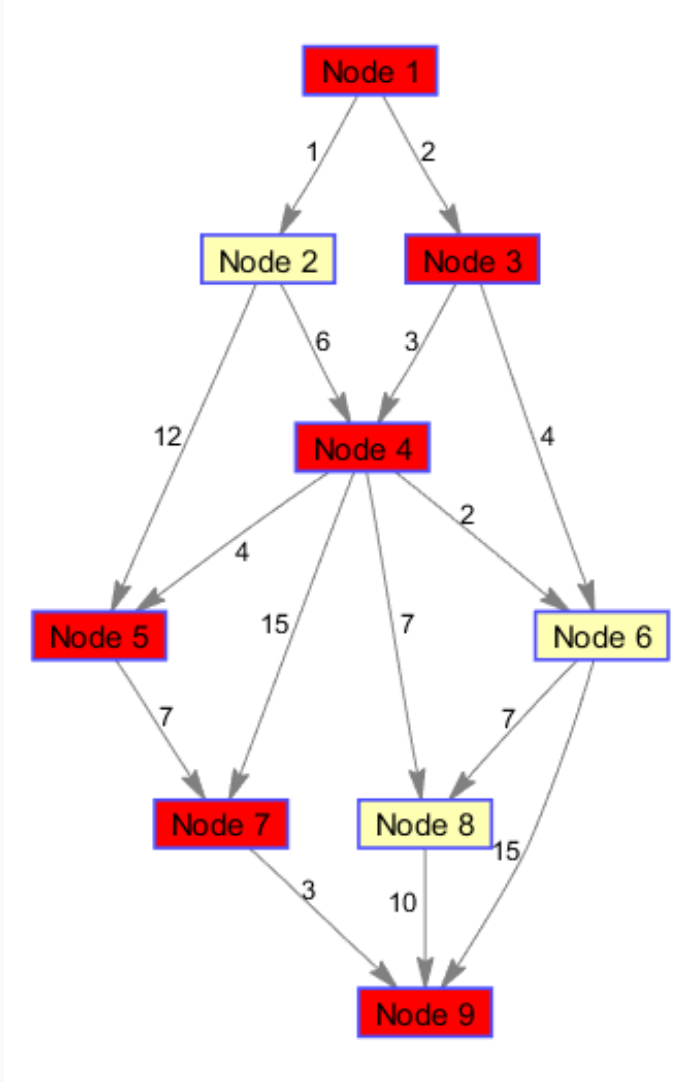


图 8: 最短路径图形显示

Dijkstra 算法 i

Dijkstra 其人：

Edsger Wybe Dijkstra (04/01/1930-08/06/2002), 荷兰皇家艺术与科学学院的院士，美国科学院院士，英国计算协会的 Fellow。年轻时代，Dijkstra 在 University of Leiden（荷兰最古老的大学）学习理论物理，但很快他就意识到其兴趣不在于理论物理虽然获得了其数学和理论物理的学位。

1984 年至 1999 年，作为计算机系系主任任职与美国 UT Austin(得克萨斯州大学奥斯汀分校)，并于 1999 年退休。

Dijkstra 算法 ii

图 9: E. W. Dijkstra

E. W. Dijkstra 15 年的学术著作覆盖了图论的理论工作，教育手册，解释文章和编程语言领域的哲学思考。

Dijkstra 算法基本思路 i

准备工作:

- 设图 G 中有 n 个顶点，设置一个集合 U 存放已求出最短路径的顶点，V-U 是尚未确定最短路径的顶点集合
- 每个顶点对应一个距离值，则集合 U 中顶点的距离值是从顶点 v0 到该顶点的最短路径长度；集合 V-U 中顶点的距离值是从顶点 v0 到该顶点的只包括集合 U 中顶点为中间顶点的最短路径长度

Dijkstra 算法基本思路 i

初始状态：

- 集合 U 中只有顶点 v0，顶点 v0 对应的距离值为 0
- 集合 V-U 中顶点 vi 的距离值为边 (v0,vi) 的权值 (i=1,⋯,n-1)，如果 v0 和 vi 间无边直接相连，则 vi 的距离值为 ∞

在集合 V-U 中选择距离值最小的顶点 vmin 加入集合 U

对集合 V-U 中各顶点的距离值进行修正，如果加入顶点 vmin 为中间顶点后，使 v0 到 vi 的距离值比原来的距离值更小，则修改 vi 的距离值

反复操作，直到从 v0 出发可以到达的所有顶点都在集合 U 中为止

Dijkstra 算法 i

```
function [d,path]=dijkstra(W,s,t)
[n,m]=size(W);ix=(W==0);W(ix)=Inf;
if n =m,error('Square W required');end
visited(1:n)=0;dist(1:n)=Inf;parent(1:n)=0;dist(s)=0;d=Inf;
for i=1:(n-1),
ix=(visited==0);vec(1:n)=Inf;vec(ix)=dist(ix);
[a,u]=min(vec);visited(u)=1;
for v=1:n
if(W(u,v)+dist(u)<dist(v)),dist(v)=dist(u)+W(u,v);parent(v)=u;
end;end;end
if parent(t) =0,path=t;d=dist(t);
```

Dijkstra 算法 ii

while t =s,p=parent(t);path=[p path];t=p;end

end

执行语句： [d p]=dijkstra(R,1,9)

返回：

d = 19

p = 1 3 4 5 7 9