最优化方法

实验指导书

**自动化系**

**2024.11**

**实验一 求解线性规划方法实验**

一、实验目的：

(1) 通过实验掌握线性规划解的几何特征；

(2) 掌握求线性规划的单纯形法、大M法。

二、实验内容：

(1) 熟悉实验设备，掌握利用MATLAB求解线性规划问题的基本步骤

例1极小化问题



例2极小化问题



例3极小化问题











例4极小化问题









1. 利用实验设备完成实际线性规划问题的计算机程序编写与仿真分析；

例5某工厂在计划期内要安排生产甲、乙两种产品，已知生产单位产品所需的设备台时及A、B两种原材料的消耗如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 甲 | 乙 | 总 |
| 设备 | 1 | 2 | 8台时 |
| A | 4 | 0 | 16kg |
| B | 0 | 4 | 12kg |

该工厂每生产一件产品甲可获利2元，每生产一件产品乙可获利3元，问应如何安排计划使该工厂获利最多？

(3) 分析实验结果，完成实验报告。  
三、实验方法、步骤及结果分析简要提示：

线性规划问题的标准形式是：



在MATLAB工具箱中，可用linprog函数求解线性规划问题。

linprog函数的调用格式如下：

●x=linprog(f,A,b)：求解问题min f \*x，约束条件为A\*x<=b。

●x=linprog(f,A,b,Aeq,beq)：求解上面的问题，但增加等式约束，即Aeq\*x=beq。若没有不等式约束，则令A=[ ],b=[ ]。

●x=linprog(f,A,b,Aeq,beq,lb,ub)：定义设计x的下界lb和上界ub，使得x始终在该范围内。若没有等式约束，令Aeq=[ ],beq=[ ]。

●x=linprog(f,A,b,Aeq,beq,lb,ub,x0)：设置初值为x0。该选项只适用于中型问题，默认时大型算法将忽略初值。

●x=linprog(f,A,b,Aeq,beq,lb,ub,x0,options)：用options指定的优化参数进行最小化。

●[x,fval]=linprog(…)：返回解x处的目标函数值fval。

●[x,lambda,exitflag]=linprog(…)：返回exitflag值，描述函数计算的退出条件。

●[x,lambda,exitflag,output]=linprog(…)：返回包含优化信息的输出参数output。

●[x,fval,exitflag,output,lambda]=linprog(…)：将解x处的拉格朗日乘子返回到lambda参数中。

调用格式中，lambda参数为解x处包含拉格朗日乘子的结构。它有以下一些字段：

lower—下界lb

upper—上界ub

ineqlin—线性不等式

eqlin—线性等式

exitflag参数表示算法终止的原因，下面列出不同值对应的退出原因：

1 函数在解x处有解

0 迭代次数超过options.MaxIter

-2 没有找到可行点

-3 问题无解

-4 执行算法时遇到NaN

-5 原问题和对偶问题都不可行

-7 搜索方向太小，不能继续前进。

例1极小化问题



>> f=[-2,-1,3,-5];

>> A=[1,2,4,-1;2,3,-1,1;1,0,1,1];

>> b=[6,12,4];

>> Aeq=[];

>> beq=[];

>> lb=[0,0,0,0];

>> ub=[];

>> [x,fval]=linprog(f,A,b,Aeq,beq,lb,ub)

输出结果：

x =[0 2.6667 0 4]

fval =-22.6667

四、实验分值：预习准备20分；实验过程40分；实验报告20分；分析思考20分。

**实验二 求解无约束非线性规划问题常见方法实验**

一、实验目的：

(1) 通过实验理解无约束非线性规划的最优性条件；

(2) 掌握常见的无约束非线性规划问题的求解方法。

二、实验内容：

(1) 应用共轭梯度法求解无约束非线性规划问题的基本步骤；

例1极小化问题，初始点，精度要求；

例2选择不同的初始点[0,0]T, [0,1]T, [1,0]T，求二次函数的极小值以及极小点，精度要求；

(2) 应用牛顿和修正牛顿法求解优化问题的基本步骤；

例3应用牛顿法极小化问题，初始点，精度要求；

例4应用牛顿法极小化问题，初始点，精度要求；

例5应用修正牛顿法或拟牛顿法求解（注：利用DFP方法及BFGS方法求解）；

(3) 利用Matlab软件求解实际优化问题；

例5最大收益问题

某商店卖两种牌子的果汁，本地牌子每瓶进价1元，外地牌子每瓶进价1.2元，店主估计，如果本地牌子的果汁每瓶卖元，外地牌子的果汁每瓶卖元，则每天可卖出瓶本地牌子的果汁，瓶外地牌子的果汁。

问1：建立商店收益模型。

问2：试用数学模型进行概括，并说明店主每天以什么价格卖两种牌子的果汁可取得最大收益。

(4) 分析实验结果，完成实验报告。

三、实验方法、步骤及结果分析简要提示：

**（1）应用共轭梯度法求解优化问题的基本步骤；**

算法步骤1：给定初值、精度要求并计算梯度和海塞矩阵，令k=0

function f=conjugate\_grad\_2d(x0,t)

x=x0; %已知初始点坐标：x0

t; %精度要求

f=f(xi,yi) %创建目标函数 f

fx=diff(f,xi); %求表达式 f 对 xi 的一阶求导

fy=diff(f,yi); %求表达式 f 对 yi 的一阶求导

fx=subs(fx,{xi,yi},x0); %代入初始点坐标计算对 xi 的一阶求导实值

fy=subs(fy,{xi,yi},x0); %代入初始点坐标计算对 yi 的一阶求导实值

fi=[fx,fy]; %初始点梯度向量

count=0; %搜索次数初始为 0

功能：给定初始条件及目标函数。

其中：diff（函数，变量）：代表对变量求偏导；subs(S,OLD,NEW) 表示将符号表达式S中的符号变量OLD替换为新的值NEW。

算法步骤2：设置终止条件

double(sqrt(fx^2+fy^2))<t； %搜索精度满足已知条件停止，否则继续迭代

功能：设置终止条件。

其中：终止条件通过向量2-范数实现。

算法步骤3：确定第一次搜索方向、步长因子及一次搜索后的迭代点

while double(sqrt(fx^2+fy^2))>t； %搜索精度不满足已知条件

s=-fi; %第一次搜索的的方向为负梯度方向

if count<=0

s=-fi;

else

s=s1;

end

x=x+a\*s; %进行一次搜索后的点坐标

f=subs(f,{xi,yi},x); %构造一元搜索的一元函数φ (a)

f1=diff(f); %对函数φ (a)进行求导

f1=solve(f1); %得到最佳步长 a

if f1~=0

ai=double(f1); %强制转换数据类型为双精度数值

else

break %若 a=0，则直接跳出循环，此点即为极值点

end

x=subs(x,a,ai); %得到一次搜索后的点坐标值

功能：一次搜索迭代点的确定。

算法步骤4：采用共轭梯度方向确定k次搜索后的迭代点

f=f(xi,yi) %创建目标函数 f

fxi=diff(f,xi);

fyi=diff(f,yi);

fxi=subs(fxi,{xi,yi},x);

fyi=subs(fyi,{xi,yi},x);

fii=[fxi,fyi]; %下一点梯度向量

d=(fxi^2+fyi^2)/(fx^2+fy^2);

s1=-fii+d\*s; %下一点搜索的方向向量

count=count+1; %搜索次数加 1

fx=fxi;

fy=fyi; %搜索后终点坐标变为下一次搜索的始点坐标

end

x,f=subs(f,{xi,yi},x),count %输出极值点，极小值以及搜索次数

功能：k次搜索迭代点的确定。

注：该算法适用于目标函数为二次函数情形，当目标函数为非二次函数时，需要在迭代n步后重新设置初始点，需要在此基础上修改。

将上述程序存成conjugate\_grad\_2d.m文件，每次在命令窗中输入初始点、精度要求并调用conjugate\_grad\_2d 函数即可。

1. 极小化问题，初始点，精度要求；

>> syms x0 t % 明确conjugate\_grad\_2d 函数中目标函数解析式后

x0=[-2,4];

t=0.001;

f=conjugate\_grad\_2d(x0,t)

1. 选择不同的初始点[0,0]T, [0,1]T, [1,0]T，求二次函数的极小值以及极小点，精度要求；

>> syms x0 t % 明确conjugate\_grad\_2d 函数中目标函数解析式后

x0=[0,0];

t=0.01;

f=conjugate\_grad\_2d(x0,t)

结果分析：

由结果看出，程序经过 2 次迭代，得到二次函数的极小值坐标[4,2],极小值-8；表明共轭梯度法收敛速度较快，计算量较小，稳定性高。

选择不同的初始点坐标,都是经过 2 次迭代得到一致的结果；表明共轭梯度法初始点的选择不影响收敛结果。

从共轭梯度法的计算过程可以看出，第一个搜索方向取作负梯度方向，这就是最速下降法。

**（2）牛顿法求解优化问题的基本步骤；**

算法步骤1：给定初值、精度要求并计算梯度和海塞矩阵，令k=0

f=f(x1,x2);

v=[x1,x2];

df=jacobian(f,v);

df=df.';

G=jacobian(df,v);

epson ;

x0=[x,y]' ;

g1=subs(df,{x1,x2},{x0(1,1),x0(2,1)});

G1=subs(G,{x1,x2},{x0(1,1),x0(2,1)});

k=0;

功能：给定初始条件及目标函数。

其中：jacobian（函数，变量）：计算函数关于对变量的雅可比矩阵；subs(S,OLD,NEW) 表示将符号表达式S中的符号变量OLD替换为新的值NEW。

算法步骤2：迭代过程

while(norm(g1)>epson)

p=-G1\g1;

x0=x0+p;

g1=subs(df,{x1,x2},{x0(1,1),x0(2,1)});

G1=subs(G,{x1,x2},{x0(1,1),x0(2,1)});

k=k+1;

end;

k

x0

功能：给出搜索方向及迭代点，满足终止条件，停止迭代。

其中：表示

例3应用牛顿法极小化问题，初始点，精度要求；

>> syms x y

f=(x-2)^2+(y-4)^2;

v=[x,y];

df=jacobian(f,v);

df=df.';

G=jacobian(df,v);

epson=0.001;

x0=[-2,4]';

g1=subs(df,{x,y},{x0(1,1),x0(2,1)});

G1=subs(G,{x,y},{x0(1,1),x0(2,1)});

k=0;

while(norm(g1)>epson)

p=-G1\g1;

x0=x0+p;

g1=subs(df,{x,y},{x0(1,1),x0(2,1)});

G1=subs(G,{x,y},{x0(1,1),x0(2,1)});

k=k+1;

end;

k

x0

**（3）拟牛顿法求解优化问题的基本步骤；**

命令行式：

x = fminunc(fun,x0)

x = fminunc(fun,x0,options)

功能：使用fminunc函数实现调用Matlab中现有的拟牛顿法的BFGS和DFP算法软件程序进行求解。

注： 1）fminunc为无约束优化提供了大型优化和中型优化算法。由options中的参数LargeScale控制：

LargeScale=’on’(默认值),使用大型算法

LargeScale=’off’(默认值),使用中型算法

2）fminunc为中型优化算法的搜索方向提供了4种算法，由options中的参数HessUpdate控制：

HessUpdate=’bfgs’（默认值），拟牛顿法的BFGS公式；

HessUpdate=’dfp’，拟牛顿法的DFP公式；

HessUpdate=’steepdesc’，最速下降法

3）fminunc为中型优化算法的步长一维搜索提供了两种算法，由options中参数LineSearchType控制：

LineSearchType=’quadcubic’(缺省值)，混合的二次和三次多项式插值；

LineSearchType=’cubicpoly’，三次多项式插

例4应用拟牛顿法求解,初始点[2,3]（注：利用DFP方法及BFGS方法求解）；

>> %建立一个m文件，输入要求解的优化问题的目标函数和梯度函数，内容如下：

function [f,g]=nnd(x)

f=100\*(x(2)-x(1)^2)^2+(1-x(1))^2;

g=[200\*(x(2)-x(1)^2)\*(-2)\*x(1)-2\*(1-x(1)),200\*(x(2)-x(1)^2)]; %其中，f是目标函数，g是梯度，然后调用optimset函数

options=optimset('LargeScale','off','HessUpdate','dfp','gradobj','on');

[x,fval]=fminunc('nnd',[2,3],options)

options=optimset('LargeScale','off','HessUpdate','bfgs','gradobj','on');

[x,fval]=fminunc('nnd',[2,3],options)

四、实验分值：预习准备20分；实验过程40分；实验报告20分；分析思考20分。

**实验三 求解约束非线性规划问题常见方法实验**

一、实验目的：

(1) 通过实验理解约束非线性规划的最优性条件；

(2) 掌握常见的约束非线性规划问题的求解方法。

二、实验内容：

(1) 二次规划问题求解步骤；





(2) 一般约束非线性规划求解步骤；





例5实例分析

某公司有6个建筑工地要开工，每个工地的位置（用平面坐标系，表示，距离单位：千米 ）及水泥日用量(吨)由下表给出。目前有两个临时料场位于A(5,1)，B(2,7)，日储量各有20吨。假设从料场到工地之间均有直线道路相连。试制定每天的供应计划，即从A，B两料场分别向各工地运送多少吨水泥，使总的吨千米数最小。

工地位置及水泥日用量

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 3 | 5 | 6 |
|  | 1.25  1.25  3 | 8.75  0.75  5 | 0.5  4.75  4 | 5.75  5  7 | 3  6.5  6 | 7.25  7.25  11 |

(3) 分析实验结果，完成实验报告。

三、实验方法、步骤及结果分析简要提示：

**（1）二次规划问题求解的基本步骤；**

将二次规划问题写成标准型



输入命令

H=[2 -2; -2 4];

c=[-2 ;-6];

A=[1 1; -1 2];b=[2;2];

Aeq=[];beq=[]; VLB=[0;0];VUB=[];

[x,z]=quadprog(H,c,A,b,Aeq,beq,VLB,VUB)

注：用Matlab软件求解,其**输入格式**如下:

1. x=quadprog(H,C,A,b);

2. x=quadprog(H,C,A,b,Aeq,beq);

3. x=quadprog(H,C,A,b,Aeq,beq,VLB,VUB);

4. x=quadprog(H,C,A,b, Aeq,beq ,VLB,VUB,X0);

5. x=quadprog(H,C,A,b, Aeq,beq ,VLB,VUB,X0,options);

6. [x,fval]=quaprog(...);

7. [x,fval,exitflag]=quaprog(...);

8. [x,fval,exitflag,output]=quaprog(...);

运行结果

x =0.8 1.2

z = -7.2

**（2）一般约束非线性规划问题求解的基本步骤；**

将约束非线性优化写成标准型



其中，为n维变元向量，与均为非线性函数组成的向量，其它变量的含义与线性规划、二次规划中相同。用Matlab求解上述问题，基本步骤分三步：

首先，建立M文件fun.m,定义目标函数F（X）:

function f=fun(X);

f=F(X);

若约束条件中有非线性约束：或,则建立M文nonlcon.m定义函数G(X)与Ceq(X):

function [G,Ceq]=nonlcon(X)

G=...

Ceq=...

其次，建立主程序。非线性规划求解的函数是fmincon命令的基本格式如下：

x=fmincon(‘fun’,X0,A,b)

x=fmincon(‘fun’,X0,A,b,Aeq,beq)

x=fmincon(‘fun’,X0,A,b,Aeq,beq,VLB,VUB) x=fmincon(‘fun’,X0,A,b,Aeq,beq,VLB,VUB,’nonlcon’)

x=fmincon(‘fun’,X0,A,b,Aeq,beq,VLB,VUB,’nonlcon’,options)

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

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

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

以如下优化问题为例：



建立m文件fun4.m 定义目标函数

function f=fun4(x);

f=exp(x(1))\*(4\*x(1)^2+2\*x(2)^2+4\*x(1)\*x(2)+2\*x(2)+1);

再建立M文件com.m定义非线性约束：

function [d e]=com(x);

d=1.5+x(1)\*x(2)-x(1)-x(2);

e=-x(1)\*x(2)-10;

建立主程序

x0=[-2;2];

A=[];b=[];

Aeq=[1 1];beq=[0];

vlb=[];vub=[];

[x,fval]=fmincon('fun4',x0,A,b,Aeq,beq,vlb,vub,'com')

输出结果

x =[-3.1623 3.1623]

fval = 1.1566

**（3）实例分析；**

记工地的位置为，水泥日用量为；料场位置为，日储量为；从料场*j*向工地*i*的运送量为。

目标函数为：

约束条件为：

四、实验分值：预习准备20分；实验过程40分；实验报告20分；分析思考20分。