OR 第九-十周上机作业

20123101 李昀哲

2023.2.25

外点罚函数法求解：

$$\boldsymbol{min\ f(x) = (x_1 - 2)^2 + (x_2 - 1)^2}$$

$$\text{s.t.}\begin{cases} -0.25x_1^2 - x_2^2 + 1 \geq 0 \\ x_1 - 2x_2 + 1 = 0 \end{cases}$$

接口函数[xstar, fxstar, iter] = penalty(penalty_func, contrains, x_0, ε)

初始迭代点$x_0 = (2,2)$，ε= 1e-3

算法流程

- 构造惩罚函数：F=f+M * { u*[g1(x)]^2 + [h2(x)]^2 } ，式中 M 为初始惩罚因子,（外点惩罚函数，迭代点再可行域之外，不等式约束才起作用）
- 然后用无约束优化极值算法求解；
- 如果相邻两次惩罚函数无约束最优点之间的距离足够小（norm(x1-x0)<eps），则收敛；否则放大惩罚因子 M=C*M，式中 C 为 罚因子放大系数；
- 转步骤 a 继续迭代；

罚函数：

```
function [xstar,fxstar, iter]=penalty(penalty_func,constrains,h,x0,eps)
% f 目标函数
% g 不等式约束函数矩阵
% h 等式约束函数矩阵
% x0 初始值
% eps 退出容差
M=0.01;% M 初始惩罚因子
C=3;% C 罚因子放大倍数
penalty = sum(h.^2);
iter = 1;
while iter
    % 判断在不在可行域内
    gx=double(subs(constrains,symvar(constrains),x0));
    index=find(gx<0);
    F_NEQ=sum(constrains(index).^2);
    F=matlabFunction(penalty_func+M*F_NEQ+M*penalty);
    x1=Min_Newton(F,x0,eps,100);
    x1=x1'
    if norm(x1-x0)<eps
        xstar=x1;
        fxstar=double(subs(penalty_func,symvar(penalty_func),xstar));
        break;
    else
        M=M*C;
        x0=x1;
    end
    iter=iter+1;
end
end
```

牛顿法 Min_Newton：

```
% Operational Research
% @author 李昀哲 20123101
% Feb 25, 2023
function [X,result]=Min_Newton(f,x0,eps,n)
%f为目标函数
%x0为初始点
%eps为迭代精度
%n为迭代次数

% 求梯度和hessian矩阵
grad = gradient(sym(f),symvar(sym(f)));
Hessian=jacobian(grad,symvar(sym(f)));
Var_grad=symvar(grad);
Var_Hessian=symvar(Hessian);
Var_Num_grad=length(Var_grad);
Var_Num_Hessian=length(Var_Hessian);
grad=matlabFunction(grad);
flag = 0;

if Var_Num_Hessian == 0  % 判断hessian矩阵是常数
    Hessian=double((Hessian));
    flag=1;
end
% 求当前点梯度与hessian矩阵的逆
f_cal='f(';
TiDu_cal='TiDu(';
Haisai_cal='Haisai(';
for k=1:length(x0)
    f_cal=[f_cal,'x0(',num2str(k),'),'];

    for j=1: Var_Num_grad
        if char(Var_grad(j)) == ['x',num2str(k)]
            TiDu_cal=[TiDu_cal,'x0(',num2str(k),'),'];
        end
    end

    for j=1:Var_Num_Hessian
        if char(Var_Hessian(j)) == ['x',num2str(k)]
            Haisai_cal=[Haisai_cal,'x0(',num2str(k),'),'];
        end
    end
end
```

```
Hessian_cal(end)=')';
Grad_cal(end)=')';
f_cal(end)=')';
switch flag
    case 0
        Hessian=matlabFunction(Hessian);
        dk='-eval(Hessian_cal)^(-1)*eval(Grad_cal)';
    case 1
        dk='-Hessian^(-1)*eval(Grad_cal)';
        Hessian_cal='Hessian';
end
i=1;
while i < n
    if abs(det(eval(Hessian_cal))) < 1e-6
        disp('逆矩阵不存在！');
        break;
    end
    x0=x0(:)+eval(dk);
    if norm(eval(Grad_cal)) < eps
        X=x0;
        result=eval(f_cal);
        return;
    end
    i=i+1;
end
disp('无法收敛！');
X=[];
result=[];
end
```

运行结果：

```
xstar =

    0.8231    0.9115


fxstar =

    1.3929


iter =

    13
```