function [x,mu,lambda,output]=multphr(fun,hf,gf,dfun,dhf,dgf,x0)

%功能: 用乘子法解一般约束问题: min f(x), s.t. h(x)=0, g(x).=0

%输入: x0是初始点, fun, dfun分别是目标函数及其梯度；

% hf, dhf分别是等式约束（向量）函数及其Jacobi矩阵的转置；

% gf, dgf分别是不等式约束（向量）函数及其Jacobi矩阵的转置；

%输出: x是近似最优点，mu, lambda分别是相应于等式约束和不

% 等式约束的乘子向量; output是结构变量, 输出近似极小值f, 迭

% 代次数, 内迭代次数等

maxk=500; %最大迭代次数

sigma=2.0; %罚因子

eta=2.0; theta=0.8; %PHR算法中的实参数

k=0; ink=0; %k, ink分别是外迭代和内迭代次数

epsilon=1e-5; %终止误差值

x=x0; he=feval(hf,x); gi=feval(gf,x);

n=length(x); l=length(he); m=length(gi);

%选取乘子向量的初始值

mu=0.1\*ones(l,1); lambda=0.1\*ones(m,1);

btak=10; btaold=10; %用来检验终止条件的两个值

while(btak>epsilon & k<maxk)

%调用BFGS算法程序求解无约束子问题

[x,ival,ik]=bfgs('mpsi','dmpsi',x0,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma);

ink=ink+ik;

he=feval(hf,x); gi=feval(gf,x);

btak=0.0;

for(i=1:l), btak=btak+he(i)^2; end

for(i=1:m)

temp=min(gi(i),lambda(i)/sigma);

btak=btak+temp^2;

end

btak=sqrt(btak);

if btak>epsilon

if(k>=2 & btak>theta\*btaold)

sigma=eta\*sigma;

end

%更新乘子向量

for(i=1:l), mu(i)=mu(i)-sigma\*he(i); end

for(i=1:m)

lambda(i)=max(0.0,lambda(i)-sigma\*gi(i));

end

end

k=k+1;

btaold=btak;

x0=x;

end

f=feval(fun,x);

output.fval=f;

output.iter=k;

output.inner\_iter=ink;

output.bta=btak;

%%%%%%%%%%%%%%%%%% 增广拉格朗日函数%%%%%%%%%%%%%%%%%%%%%

function psi=mpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma)

f=feval(fun,x); he=feval(hf,x); gi=feval(gf,x);

l=length(he); m=length(gi);

psi=f; s1=0.0;

for(i=1:l)

psi=psi-he(i)\*mu(i);

s1=s1+he(i)^2;

end

psi=psi+0.5\*sigma\*s1;

s2=0.0;

for(i=1:m)

s3=max(0.0, lambda(i) - sigma\*gi(i));

s2=s2+s3^2-lambda(i)^2;

end

psi=psi+s2/(2.0\*sigma);

%%%%%%%%%%%%%%%%%% 增广拉格朗日函数的梯度%%%%%%%%%%%%%%%%%%%%

function dpsi=dmpsi(x,fun,hf,gf,dfun,dhf,dgf,mu,lambda,sigma)

dpsi=feval(dfun,x);

he=feval(hf,x); gi=feval(gf,x);

dhe=feval(dhf,x); dgi=feval(dgf,x);

l=length(he); m=length(gi);

for(i=1:l)

dpsi=dpsi+(sigma\*he(i)-mu(i))\*dhe(:,i);

end

for(i=1:m)

dpsi=dpsi+(sigma\*gi(i)-lambda(i))\*dgi(:,i);

end

function [x,val,k]=bfgs(fun,gfun,x0,varargin)

%功能: 用BFGS算法求解无约束问题: min f(x)

%输入: x0是初始点, fun, gfun分别是目标函数及其梯度;

% varargin是输入的可变参数变量, 简单调用bfgs时可以忽略它,

% 但若其它程序循环调用该程序时将发挥重要的作用

%输出: x, val分别是近似最优点和最优值, k是迭代次数.

maxk=500; %给出最大迭代次数

rho=0.55;sigma=0.4; epsilon=1e-5;

k=0; n=length(x0);

Bk=eye(n); %Bk=feval('Hess',x0);

while(k.maxk)

gk=feval(gfun,x0,varargin{:}); %计算梯度

if(norm(gk).epsilon), break; end %检验终止准则

dk=-Bk“gk; %解方程组, 计算搜索方向

m=0; mk=0;

while(m.20) % 用Armijo搜索求步长

newf=feval(fun,x0+rho^m\*dk,varargin{:});

oldf=feval(fun,x0,varargin{:});

if(newf.oldf+sigma\*rho^m\*gk'\*dk)

mk=m; break;

end

m=m+1;

end

%BFGS校正

x=x0+rho^mk\*dk;

sk=x-x0; yk=feval(gfun,x,varargin{:})-gk;

if(yk’\*sk.0)

Bk=Bk-(Bk\*sk\*sk'\*Bk)/(sk'\*Bk\*sk)+(yk\*yk')/(yk'\*sk);

end

k=k+1; x0=x;

end

val=feval(fun,x0,varargin{:});