% Course: Nonlinear Optimization. %

% FALL.2018. Dr. Cheng. %

% Assignment: (7) %

% Date:(2018.10.24) %

% By: 卢博 %

% ID NUMBER: 11849159 %

% LAB:(7) %

%

1. Multiple-Run Gradient Search

clear;clc  
syms x y  
f = (1 - x)^2+100\*(y - x^2)^2;  
[x,k] = grad(f,-2,2)

x =  
 1.0000  
 1.0000  
k =  
 25212

function [xout,k] = grad(f,lb,ub)  
g=matlabFunction(gradient(f));  
x = lb:.01:ub;  
y = lb:.01:ub;  
index = randi(length(x));  
x\_best = x(index);  
y\_best = y(index);  
a = x\_best ;  
b = y\_best;  
u = [a b]';  
h = 0.001;  
m = -2;  
n = 0;  
k = 0;  
while abs(m) > 1.0e-5 || abs(n) > 1.0e-5  
 A = [m n]';  
 u = u - h\*A;  
 if ismember(1,u<lb) || ismember(1,u>ub)  
 break;  
 end  
 a = u(1);  
 b = u(2);  
 S = g(a,b);  
 m = S(1);  
 n = S(2);  
 k = k + 1;  
end  
xout = u;  
end

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

1. Random Search

clear;clc;  
f = @(x,y) (1-x).^2 + 100\*(y - x.^2).^2;  
lb = -2;ub = 2;  
[x,y,f,k]=rsa(f,lb,ub)

x =  
 0.9951  
y =  
 0.9894  
f =  
 9.3499e-05  
k =  
 9830

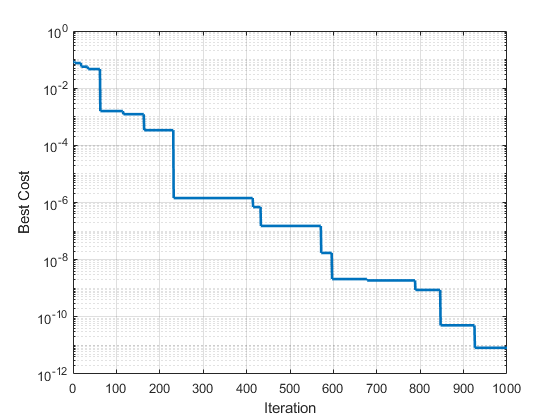
function [x\_best,y\_best,fout,knum] = rsa(f,lb,ub)  
x = lb:.01:ub;  
y = lb:.01:ub;  
  
x\_best = x(randi(length(x)));  
y\_best = y(randi(length(x)));  
f\_best = f(x\_best,y\_best);  
k = 0;k\_max = 10000;  
lambda = 1;  
while true  
 x\_new = x(randi(length(x)));  
 y\_new = y(randi(length(x)));  
 x\_new = lambda\*x\_new + (1-lambda)\*x\_best;  
 y\_new = lambda\*y\_new + (1-lambda)\*y\_best;  
 f\_new = f(x\_new,y\_new);  
 if ismember(1,[x\_new y\_new]<lb) || ismember(1,[x\_new y\_new]>ub)  
 fout = f\_best;  
 knum = k;  
 break;  
 end  
 if f\_new < f\_best  
 delta = f\_best - f\_new;  
 x\_best = x\_new;  
 y\_best = y\_new;  
 f\_best = f\_new;  
 if delta < 1e-4  
 fout = f\_best;  
 knum = k+1;  
 break;  
 end  
 end  
 lambda = (k\_max-k)/k\_max;  
 k = k + 1;  
end  
end

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

1. Differential Evolution Algorithm

clear;clc  
f = @(x) (1-x(1)).^2 + 100\*(x(2) - x(1).^2).^2;  
dea(f)

Iteration 1000: Best Cost = 8.2314e-12



Problem Definition 1

DE Parameters 1

Initialization 1

DE Main Loop 2

Show Results 3

function dea(fun)

## Problem Definition

CostFunction= fun; % Cost Function  
nVar=20; % Number of Decision Variables  
VarSize=[1 nVar]; % Decision Variables Matrix Size  
VarMin=-2; % Lower Bound of Decision Variables  
VarMax= 2; % Upper Bound of Decision Variables

## DE Parameters

MaxIt=1000; % Maximum Number of Iterations  
nPop=50; % Population Size  
beta\_min=0.2; % Lower Bound of Scaling Factor  
beta\_max=0.8; % Upper Bound of Scaling Factor  
pCR=0.2; % Crossover Probability

## Initialization

empty\_individual.Position=[];  
empty\_individual.Cost=[];  
BestSol.Cost=inf;  
pop=repmat(empty\_individual,nPop,1);  
for i=1:nPop  
 pop(i).Position=unifrnd(VarMin,VarMax,VarSize);  
 pop(i).Cost=CostFunction(pop(i).Position);  
 if pop(i).Cost<BestSol.Cost  
 BestSol=pop(i);  
 end  
end  
BestCost=zeros(MaxIt,1);

## DE Main Loop

for it=1:MaxIt  
 for i=1:nPop  
 x=pop(i).Position;  
 A=randperm(nPop);  
 A(A==i)=[];  
 a=A(1);  
 b=A(2);  
 c=A(3);  
 % Mutation  
 %beta=unifrnd(beta\_min,beta\_max);  
 beta=unifrnd(beta\_min,beta\_max,VarSize);  
 y=pop(a).Position+beta.\*(pop(b).Position-pop(c).Position);  
 y = max(y, VarMin);  
 y = min(y, VarMax);  
 % Crossover  
 z=zeros(size(x));  
 j0=randi([1 numel(x)]);  
 for j=1:numel(x)  
 if j==j0 || rand<=pCR  
 z(j)=y(j);  
 else  
 z(j)=x(j);  
 end  
 end  
 NewSol.Position=z;  
 NewSol.Cost=CostFunction(NewSol.Position);  
 if NewSol.Cost<pop(i).Cost  
 pop(i)=NewSol;  
 if pop(i).Cost<BestSol.Cost  
 BestSol=pop(i);  
 end  
 end  
 end  
 % Update Best Cost  
 BestCost(it)=BestSol.Cost;  
 % Show Iteration Information  
% disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);  
end

## Show Results

disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);  
figure;  
%plot(BestCost);  
semilogy(BestCost, 'LineWidth', 2);  
xlabel('Iteration');  
ylabel('Best Cost');  
grid on;

end

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)