

الـكـورـيـتم DE

```
clc;
clear all;
close all;

%% Defining problem
CostFunction=@(x) sphere(x);
nvar=10;
varSize=[1 nvar];
var_min=-5;
var_max=5;

%% Defining DE parameters
MaxIt=100;
nPop=100;
beta=0.5;
% beta_max=0.8;
% beta_min=0.2;
Pcr=.3;

%% initialization
Individual.position=[];
Individual.cost=[];
Pop= repmat(Individual,nPop,1);

BestSol.cost=inf;

for i=1:nPop
    Pop(i).position=unifrnd(var_min,var_max,varSize);
    Pop(i).cost=CostFunction(Pop(i).position);
    if Pop(i).cost<BestSol.cost
        BestSol=Pop(i);
    end
end

BestCost=zeros(1,MaxIt);
```

```

%% 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,varSize);
        %         beta=unifrnd(beta_min,beta_max);
        y=Pop(a).position+beta.*(Pop(b).position-Pop(c).position);
        y=max(y, var_min);
        y=min(y, var_max);
        %----Crossover-----
        z=zeros(size(x));
        j0=randi([1,nvar]);
        for j=1:nvar
            if j==j0 || rand<=Pcr
                z(j)=y(j);
            else
                z(j)=x(j);
            end
        end
        %----selection-----
        NewSol.position=z;
        NewSol.cost=CostFunction(z);
        if NewSol.cost<Pop(i).cost
            Pop(i)=NewSol;
            if Pop(i).cost<BestSol.cost
                BestSol=Pop(i);
            end
        end
    end
    BestCost(it)=BestSol.cost;
end

```

```
%----Show Results-----  
disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))])  
end  
figure  
plot(BestCost)  
% semilogy(BestCost)  
BestSol.position
```

تابع هدف/ هزینه مسئله (کره)

```
function z=sphere(x)  
z=sum(x.^2);  
end
```

الکوریتم DE کوله پستی

```
clc;
clear all;
close all;

%% Defining problem
CostFunction=@(xhat) Knapsack(xhat);
nvar=10;
varSize=[1 nvar];
var_min=0;
var_max=1;

%% Defining DE parameters
MaxIt=40;
nPop=20;
% beta=0.5;
beta_max=0.8;
beta_min=0.2;
Pcr=.3;

%% initialization
Individual.position=[];
Individual.cost=[];
Individual.Sol=[];

Pop= repmat(Individual,nPop,1);

BestSol.cost=inf;

for i=1:nPop
    Pop(i).position=unifrnd(var_min,var_max,varSize);
    [Pop(i).cost,Pop(i).Sol]=CostFunction(Pop(i).position);
    if Pop(i).cost<BestSol.cost
        BestSol=Pop(i);
    end
end
end
```

```

BestCost=zeros(1,MaxIt);

%% 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,varSize);
%         beta=unifrnd(beta_min,beta_max);
        y=Pop(a).position+beta.*(Pop(b).position-Pop(c).position);
        y=max(y, var_min);
        y=min(y, var_max);
        %----Crossover-----
        z=zeros(size(x));
        j0=randi([1,nvar]);
        for j=1:nvar
            if j==j0 || rand<=Pcr
                z(j)=y(j);
            else
                z(j)=x(j);
            end
        end
        %----selection-----
        NewSol.position=z;
        [NewSol.cost,NewSol.Sol]=CostFunction(z);
        if NewSol.cost<Pop(i).cost
            Pop(i)=NewSol;
            if Pop(i).cost<BestSol.cost
                BestSol=Pop(i);
            end
        end
    end
end
end

```

```

BestCost(it)=BestSol.cost;

%-----Show Results-----

%    disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))])

    if BestSol.Sol.IsFeasible
        FLAG=' *';
    else
        FLAG=[' Viol. = ' num2str(BestSol.Sol.Violation)];
    end
    disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it)) FLAG]);

end
figure
plot(BestCost)
% semilogy(BestCost)
BestSol.position
BestSol.Sol

```

تابع هدف مسئله (کوله پشتی)

```

function [z,Sol]=Knapsack(xhat)
alpha=100;
x=round(xhat);
V=[10 20 25 15 30 25 40 10 14 22];
W=[26 59 59 39 52 25 37 57 52 59];
W0=240;
Violation=max(sum(x.*W)/W0-1,0);
z=sum(-x.*V)+alpha*Violation;

%-----information-----
Sol.x=x;
Sol.cost=z;
Sol.Violation=Violation;
Sol.IsFeasible=(Violation==0);
end

```