```
Clc پاک کردن صفحه
Clear پاک کردن حافظه
close all  بستن پنجرهای باز
format ShortG نمایش اعداد تا ۴ رقم اعشار

%% Parameters Setting

data=load('data.mat');
فراخوانی فایل دیتا که در فانکشن کریت دیتا ساخته شده است
nvar=data.nvar;  تعداد متغیر
lb=0*ones(1,nvar);     % Lower Bound حد پایین متغیرها
ub=1*ones(1,nvar);     % Upper Bound حد بالای متغیرها


npop=40;          % Population Size
تعداد جمعیت .برای بدست آمدن بهترین نتیجه این پارامتر باید
تنظیم شود

maxiter=200;       % Maximum Number of itererations
  حداکثر تعداد تکرار .برای بدست آمدن بهترین نتیجه این
پارامتر باید تنظیم شود

pc=0.9;         % percent of crossover
درصد تقاطع.برای بدست آمدن بهترین نتیجه این پارامتر باید
تنظیم شود

ncross=2*round(npop*pc/2);  % number of crossover offspring
تعداد فرزند که به روش تقاطع بدست می آید

pm=1-pc;          %  percent of mutation
درصد جهش.برای بدست آمدن بهترین نتیجه این پارامتر باید تنظیم
شود

nmut=round(npop*pm);  % number of mutation offspring
تعداد فرزند که به روش جهش بدست می آید


data.npop=npop;
data.ncross=ncross;
data.nmut=nmut;
data.maxiter=maxiter;
data.lb=lb;data.ub=ub;
                                        ذخیره سازی اطلاعات در متغیر دیتا

%% Create Random Pop
```

```
Tic شروع زمان گیری اجرای کد
emp.x=[];
emp.fit=[];
emp.info=[];
emp.SCH=[];
pop=repmat(emp,npop,1);
ساخت جمعیت اولیه به صورت خالی که در مرحله بعدی پر میشود

for i=1:npop
    pop(i).x=unifrnd(lb,ub);
                lb , ub تولید هر جواب به صورت تصادفی پیوسته مابین

    pop(i)=fitness(pop(i),data);
    در این خط جواب اولیه تصادفی تولید شده برای محاسبه تابع هدف
وارد فانکشن فیتنس میشود که در فایل فیتنس توضیحات آن داده شده است


end



% Best Solution
[~,ind]=min([pop.fit]);
gpop=pop(ind);
شناسایی بهترین جواب

%% main loop
BEST=zeros(maxiter,1);
MEAN=zeros(maxiter,1);
 ماتریس BEST MEAN برای ذخیره سازی نتایج در این جا به صورت ٠
                تولید میشود و بعدا در حلقه اصلی پر میشود

for iter=1:maxiter به ازای هر نسل

    % crossover
    crosspop=repmat(emp,ncross,1);
ابتدا جمعیت فرزندان تقاطعی به صورت خالی تولید میشود
    crosspop=crossover(crosspop,pop,data);
فرزندان تقاطعی در این فانکشن تولید میشوند که در فایل مربوطه
توضیح آن داده شده است

     % mutation
     mutpop=repmat(emp,nmut,1);
ابتدا جمعیت فرزندان جهشی به صورت خالی تولید میشود
```

```
    mutpop=mutation(mutpop,pop,data);
```
فرزندان جهشی در این فانکشن تولید میشوند که در فایل مربوطه توضیح آن داده شده است

```
    % Merged
    [pop]=[pop;crosspop;mutpop];
```
جمعیت نسل قبل فرزندان تقاطعی فرزندان جهشی در کنار هم قرار میگیرید

```
    % Sorting
    [value,index]=sort([pop.fit]);pop=pop(index);
```
مرتب میشن از خوب به بد
```
    gpop=pop(1);
```
بهترین شناسایی میشه

```
    % Select
    pop=pop(1:npop);
```
بهترین ها انتخاب میشن و به نسل بعد میروند

```
    BEST(iter)=gpop.fit;
```
ذخیره سازی بهترین برازندگی
```
    MEAN(iter)=mean([pop.fit]);
```
ذخیره سازی میانگین برازندگی

```
    NO=' Feasible';
    if gpop.SCH>0
        NO=' Infeasible';
    end
```
در این بخش وضعیت موجه یا ناموجه بودن بهترین جواب استخراج میشود
```
    disp([ 'Iter = ' num2str(iter) ' BEST = '
num2str(BEST(iter)) NO ])
```

نمایش نتیجه این تکرار برای شماره تکرار و بهترین جواب یافت شده در این تکرار و وضعیت موجه بودن جواب

```
end

%%  Results
disp([ ' Best Fitness = '  num2str(gpop.fit)])
```
نمایش بهترین برازندگی
```
disp([ ' Time = '  num2str(toc)])
```
نمایش زمان اجرا

```
figure(1)
semilogy(BEST,'r')
```
رسم نمودار بهترین

```
hold on
semilogy(MEAN,'b')
```
رسم نمودار میانگین

```
xlabel(' iteration ')
```
لیبل زدن محور ایکس
```
ylabel(' fitness')
```
لیبل زدن محور ایگرگ
```
legend( 'BEST','MEAN')
```
لیبل زدن نمودار

```
title('GA')
```
عنوان نمودار

```
RES(gpop,data)
```
در این بخش وارد فانکشن RES میشود تا نتایج نهایی را نمایش دهد که در فایل مربوط به این فانکشن توضیحات داده شده است