❑ مسئله 04-01:

مسئله زیر را با استفاده از یک مدل الگـوریتم ژنتیـک، در فضـای اعـداد حقیقی حل کنید.

$$Min\ f(x) = 100 \times \left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$$

$$St: \quad x_1 + x_2^2 \geq 0$$

$$x_1^2 + x_2 \geq 0$$

$$-5.0 \leq x_1 \leq 5.0$$

$$-1.0 \leq x_2 \leq 1.0$$

S. M. Ashrafi, Shahid Chamran University of Ahvaz, Feb 2018

۱- حل مسئله در محیط نرم افزار متلب انجام میشود و در ابتدا مشخصات اولیه از جمله ظرفیت حافظه ، تعداد تکرار ، تعداد متغیر ها ، نرخ ترکیب ، نرخ جهش ، تعداد فرزندان ، تعداد افراد مورد جهش ، تعداد جمعیت نهایی را تعیین مینماییم.

```
%% Genetic Algorithms by constraint of homework3-1  "Eissa ravankhah asl"
40164503
clear;
close all
clc;
%% parameters setting:
popsize=1000;                          %  size of population
maxiteration=100;                      %  max of iteration
nvar=2;                                %  number of variable
cr=0.8;                                %  crossover rate
mr=0.5;                                %  mutation rate
n_child=popsize*cr;                    %  number of child
n_mutated=0.2*popsize;                 %  number of mutated
n_finalpop=popsize+n_child+n_mutated; %  number of final population
```

۲- ابعاد اولیه ماتریس پارامتر های مسئله را جهت عدم کاهش سرعت محاسبات تشکیل میدهیم.

```
%% formation about the size of the parameter matrix
pop1=nan(popsize,nvar);
child=nan(n_child,nvar);
fitness=nan(1,popsize);
c=nan(1,popsize);
```

```
prob=nan(1,popsize);
prob2=nan(1,(popsize+n_child));
baby1=nan(1,nvar);
baby2=nan(1,nvar);
mutated=nan(n_mutated,nvar);
fit=nan(1,n_finalpop);
newpop=nan(popsize,nvar);
constraint1=nan(1,8);
constraint2=nan(1,8);
viol1=nan(1,8);
viol2=nan(1,8);
m_viol=nan(1,8);
```

۳- تشکیل جمعیت اولیه با تعداد ۱۰۰۰کرومزوم و تعداد متغیر ۲ ژن که فضای متغیری برای ژن اول ۵- تا ۵+ وبرای متغیر دوم ۱- تا ۱+ در فضای پیوسته اعداد حقیقی میباشد.

```
%%  formation the initial population:
for i=1:popsize
    pop1(i,1)=-5+10*rand(1);
    pop1(i,2)=-1+2*rand(1);
end
```

۴- تشکیل حلقه اصلی فضای حل

```
%% optimization
for iter=1:maxiteration
```

۵- برازش جمعیت اولیه با استفاده از رابطه صورت مسئله

```
%%    initial population fit by problem definiton
    for i=1:popsize
        fitness(i)=100*(pop1(i,2)-pop1(i,1)^2)^2+(1-pop1(i,1))^2;
    end
```

۶- محاسبه احتمال انتخاب والدین نسل بعد بدین شکل که کرومزوم هایی که فیتنس کمتری دارند احتمال انتخاب به عنوان والدین نسل بعد و انتقال ویژگی های خوبشان به نسل بعد را بیشتر دارند.

```
    %% calculate the probability of selection of parents
     for i=1:popsize
        c(i)=abs(1/(1+fitness(i)));
    end
    for i=1:popsize
        prob(i)=(c(i)/sum(c));
    end
```

۷- انتخاب والدین با استفاده از چرخ گردون(دستور randsrc) و احتمال محاسبه شده در بخش بالا و استفاده از ترکیب ارتمتیک جهت ایجاد فرزندان نسل بعد . جهت اکسپلوتیشن بهتر وفراتر رفتن فرزند از محدوده والدین بردار تصادفی بین ۰و۱ با ۵٪ درصد تلورانس از بالا و پایین محاسبه میشود یعنی(۰/۰۵- تا ۱/۰۵+)

```
%% arithmetic crossover
   for k=1:2:popsize
       % parent selection
        parentnum1=randsrc(1,1,[1:popsize;prob]);
        parentnum2=randsrc(1,1,[1:popsize;prob]);
        parent1=pop1(parentnum1,:);
        parent2=pop1(parentnum2,:);
        gen=-0.05+1.1*rand(1,nvar);
            for i=1:nvar
                baby1(i)=(gen(i)*parent1(i))+((1-gen(i))*parent2(i));
                baby2(i)=(gen(i)*parent2(i))+((1-gen(i))*parent1(i));
            end
            child(k,:)=baby1;
            child(k+1,:)=baby2;
   end
```

۸- تشکیل جمعیت ثانویه متشکل از والیدن و فرزندان

```
   %% formation of a new population
   pop2=[pop1;child];
   pop2_size=numel(pop2(:,1));
```

۹- برازش جمعیت ثانویه با استفاده از رابطه صورت مسئله

```
%%  pop2 fit by problem definiton
    for i=1:pop2_size
        fitness(i)=100*(pop2(i,2)-pop2(i,1)^2)^2+(1-pop2(i,1))^2;
    end
```

۱۰- محاسبه احتمال انتخاب جمعیت جهت اعمال جهش بدین شکل که افرادی که فیتنس بیشتری (نامطلوب تری) دارند احتمال بیشتری برای انتخاب شدن دارند تا بتوان از جمعیت نامطلوب با اعمال جهش احتمال به وجود آمدن جمعیت مطلوب را افزایش داد.

```
  %% calculate the probability of  selection of mutation population
    for i=1:pop2_size
        prob2(i)=abs(fitness(i))/sum(abs(fitness));
    end
```

۱۱- انتخاب جمعیت جهت اعمال جهش با استفاده از چرخ گردون(دستور randsrc) و اعمال جهش بدین شکل است که ژن انتخاب شده به تصادف یک در صد کاهش یا افزایش میابد. درصد یک از تحلیل حساسیت مسئله بدست آمده است و صرفا برای این مسئله کاربرد دارد.

```
%%   mutation
        for k=1:n_mutated
            mutated_num=randsrc(1,1,[1:pop2_size;prob2]);
            mutated(k,:)=pop2(mutated_num,:);
                    mutated_gen=randsrc(1,(mr*nvar),1:nvar);
                    for x=1:(mr*nvar)
```

```
mutated(k,mutated_gen(x))=(0.99+(0.02)*rand)*mutated(k,mutated_gen(x));
                    end
```

۱۲- تشکیل جمعیت نهایی از جمعیت اولیه ، فرزندان و جمعیت جهش یافته

```
%% formation the final population
finalpop=[pop1;child;mutated];
```

۱۳- برازش جعیت نهایی

```
%% new population assissment
     for i=1:n_finalpop
   fit(i)=100*(finalpop(i,2)-finalpop(i,1)^2)^2+(1-finalpop(i,1))^2;
     end
```

۱۴- اعمال قیود و محاسبه تخطی متغیر ها از قیود و اعمال تابع جریمه بدین شکل که فیتنس کروموزوم متخطی را به شکلی افزایش داده تا در ترانکیت مرحله بعد خذف شود و جواب های نامطلوب از حافظه خارج شود.

```
%% constraints
     for i=1:n_finalpop
       constraint1(i)=finalpop(i,1)+(finalpop(i,2))^2;
       constraint2(i)=(finalpop(i,1))^2+finalpop(i,2);
       v_constaint1=eps;
       v_constaint2=eps;
        viol1(i)=max(1-((constraint1(i)+0.1)/(v_constaint1)),0);
        viol2(i)=max(1-((constraint2(i)+0.1)/(v_constaint2)),0);
        m_viol(i)=((viol1(i)+ viol2(i))/2);
        alpha=10;
        fit(i)=fit(i)+alpha*viol1(i)*fit(i)+alpha*viol2(i)*fit(i);
     end
```

۱۵-تعیین بهترین فرد از جمعیت از نظر فیتنس و ترانکیت کردن جمعیت به ابعاد حافظه

```
[val,idx]=sort(fit);
      for i=1:popsize
         newpop(i,:)=finalpop(idx(i),:);
      end
      bestfit=val(1);
```

۱۶- رسم نمودار همگرایی ، (دستور pause) جهت اجرای انیمیشن وار نمودار

```
     plot(iter,bestfit,'k. ');
    hold on;
   %pause(0.0000000001);
```

۱۷- و در انتها تشیکل مجدد جمعیت اولیه جهت اجرای ایتریشن بعد و اتمام حلقه اصلی حل

```
    pop1=newpop;
end
```

۱۷- در انتهای کد خروجی های مورد نظر ایجاد میشود.

```
best_pop=finalpop(idx(1),:);
disp('best pop=');
disp(best_pop);
```

```
disp('best fit');
disp(bestfit);
%% export
 csvwrite('Export.csv',best_pop);
```

۱۸- نتایج کد نویسی

0.91039 0.82836

بعد از ۱۰۰ ایتریشن در نهایت متغیر ها برابر با جدول خروجی اکسل فوق شده و مینیمم رابطه صورت مسئله برابر با ۰/۰۰۸۱ میشود که به جواب اصلی ریاضی که ۰ است نزدیک و عملکرد الگوریتم قابل قبول میشود.