智慧型系統概論\_G.A. & E.A.

0056C203 廖育賢

# objective - the problem and the purpose

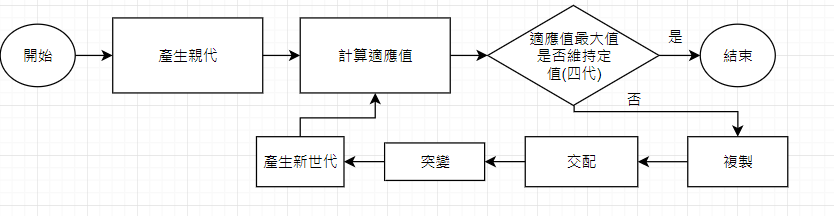
找出f(x)=- 15\*sin(2\*x)^2-(x-2)^2+160 之最大值。

# procedure

## method

以生物進化的角度，從複製、交配到突變，將不能適應環境的基因淘汰，再將存活下來的基因進行演化，將此觀點帶入至我們的問題，找出最佳解。

## program flow chart



## equation

### binary-code GA

複製: (1) 輪盤法:將親代N人依照適應值做圓餅圖，連續取N人 (2) 競爭法:從親代N人中隨機取出兩個或多個候選，適應值最高者存活，重複N次

交配: (1) one-point:選出一個位元進行交換 (2) two-point:選出兩位元，分別當作首尾，經過的位元皆交換 (3) mask:隨機產生一組序列(binary)，若為’1’則交換，若為’0’不交換

突變: (1) one-bit mutation (2) string mutation (3) uniform(mask) mutation

### real-valued GA

交配: x1’=x1+σ(x1+x2), x2’=x2-σ(x1-x2)，其中 -1<σ<1，若σ>0則x1’與x2’遠離，若σ<0則x1’與x2’靠近

突變: x2=x1+s\*random\_noise，s>0，random\_noise ∈ [-1,1]

### ea

交配: (1) Average :x=1/2\*(xj+xk) (2) Convex combination : x=r \*xj+(1-r)\*xk，r∈ (0,1)

突變: x2=x1+r\*d，r ∈ (0,1)，d為任意實數

# simulation results

## program codes

%%

%Binary code GA

clear,clc;

reproduction=1; %1:r.w.select. 2: t.select

crossover=2;%1:one-point 2: two-point

q=3;

X=zeros(1,q);

Y=zeros(1,q);

for e=1:q %執行次數

k=0; %世代

r = randi([0 2^10-1],10,1);

b = de2bi(r,10, 'left-msb');

G\_temp=[ r b];

while 1

x(1)=0;

for i=1:10

f\_fit(i)=round(1\*f(y(G\_temp(i,1)))^4);

G(k+1,i)= y(G\_temp(i,1));%x

G(k+1,i+10)=f(y(G\_temp(i,1)));%f(x)

G(k+1,i+20)=f\_fit(i);

x(i+1)=x(i)+f\_fit(i);

end

G(k+1,31)=max(G(k+1,21:30));%f\_fit max

G(k+1,32)=x(11)/10;%f\_fit average

if k>50

if G(k+1,31)==G(k,31) && G(k,31)==G(k-1,31) && G(k-1,31)==G(k-2,31)%

figure,

Y(e)=max(G(k+1,11:20));

o=find(G(k+1,11:20)==max(G(k+1,11:20)));

X(e)=G(k+1,o(1));

plot(0:k,G(1:k+1,31));

hold on;

plot(0:k,G(1:k+1,32),'--');

title(['(' num2str(X(e)) ' , ' num2str(Y(e)) ')']);

break;

end

end

if reproduction==1

%roulette wheel selection

for i=1:10

r=rand;

if r>=x(1)/x(11) && r<=x(2)/x(11)

rw=G\_temp(1,1);

elseif r>x(2)/x(11) && r<=x(3)/x(11)

rw=G\_temp(2,1);

elseif r>x(3)/x(11) && r<=x(4)/x(11)

rw=G\_temp(3,1);

elseif r>x(4)/x(11) && r<=x(5)/x(11)

rw=G\_temp(4,1);

elseif r>x(5)/x(11) && r<=x(6)/x(11)

rw=G\_temp(5,1);

elseif r>x(6)/x(11) && r<=x(7)/x(11)

rw=G\_temp(6,1);

elseif r>x(7)/x(11) && r<=x(8)/x(11)

rw=G\_temp(7,1);

elseif r>x(8)/x(11) && r<=x(9)/x(11)

rw=G\_temp(8,1);

elseif r>x(9)/x(11) && r<=x(10)/x(11)

rw=G\_temp(9,1);

elseif r>x(10)/x(11) && r<=x(11)/x(11)

rw=G\_temp(10,1);

end

G\_temp(i,:)=[ rw de2bi(rw,10, 'left-msb')];

end

else

%tournament selection 抓2

temp=zeros(10,11);

for i=1:10

r=randperm(10,2);

t=[f\_fit(r(1)),f\_fit(r(2))] ;

T=find(t==max(t));

temp(i,:)=[ G\_temp(r(T(1)),1) de2bi(G\_temp(r(T(1)),1),10, 'left-msb')];

end

G\_temp=temp;

end

%crossover(rate=0.8)

G\_cr=G\_temp;

if crossover==1 %one-point

for i=1:2:7

j = randperm(10,1);

G\_cr(i,j+1)=G\_temp(i+1,j+1);

G\_cr(i+1,j+1)=G\_temp(i,j+1);

end

else%two-point

for i=1:2:7

r = randperm(10,2);%不重複之1~10值2個

if r(1)>r(2)

temp=r(1);

r(1)=r(2);

r(2)=temp;

end

for j=r(1):r(2)

G\_cr(i,j+1)=G\_temp(i+1,j+1);

G\_cr(i+1,j+1)=G\_temp(i,j+1);;

end

end

end

for i=1:10

G\_cr(i,1)=bin2dec(strcat(int2str(G\_cr(i,2:11))));

%G\_cr(i,2:11)=de2bi(G\_cr(i,1), 10,'left-msb');

end

%mutation (rate=0.01)

if k~=0 && mod(k,1)==0

r = randperm(10,2);%產生不重複1~10之值2個

if G\_cr(r(1),r(2)+1)==1

G\_cr(r(1),r(2)+1)=0;

else

G\_cr(r(1),r(2)+1)=1;

end

G\_cr(r(1),1)=bin2dec(strcat(int2str(G\_cr(r(1),2:11))));

%G\_cr(r(1),2:11)=de2bi(G\_cr(r(1),1), 10,'left-msb');

G\_temp=G\_cr;

end

k=k+1;

end

end

figure,subplot(211),

stem(X(:),Y(:),'filled');

axis([-10 10 0 165]);

grid on

h=find(Y==max(Y))

title(['(' num2str(X(h(1))) ' , ' num2str(max(Y(:))) ')']);

uu=1;

for u=-10:0.1:10

U(uu)=f(u);

uu=uu+1

end

subplot(212),stem(-10:0.1:10,U);

hold on,grid on

stem(X(h(1)),max(Y(:)),'filled','r');

%%

%Real valued GA

clear,clc;

reproduction=1; %1:r.w.select. 2: t.select

s=5;%mutation factor

q=5;

X=zeros(1,q);

Y=zeros(1,q);

for e=1:q %執行次數

k=0;%世代

r = rand(10,1)\*20-10;% -10~10

G\_temp=[ r ];

while 1

x(1)=0;

for i=1:10

f\_fit(i)=100\* f(G\_temp(i,1))^2;

G(k+1,i)= (G\_temp(i,1));%x

G(k+1,i+10)=f(G\_temp(i,1));%f(x)

G(k+1,i+20)=f\_fit(i);

x(i+1)=x(i)+f\_fit(i);

end

G(k+1,31)=max(G(k+1,21:30));%f\_fit max

G(k+1,32)=x(11)/10;%f\_fit average

if k>50

if G(k+1,31)==G(k,31) && G(k,31)==G(k-1,31) && G(k-1,31)==G(k-2,31)

figure,

Y(e)=max(G(k+1,11:20));

o=find(G(k+1,11:20)==max(G(k+1,11:20)));

X(e)=G(k+1,o(1));

plot(0:k,G(1:k+1,31));

hold on;

plot(0:k,G(1:k+1,32),'--');

title(['(' num2str(X(e)) ' , ' num2str(Y(e)) ')']);

break;

end

end

if reproduction==1

%roulette wheel selection

for i=1:10

r=rand;

if r>=x(1)/x(11) && r<=x(2)/x(11)

rw=G\_temp(1,1);

elseif r>x(2)/x(11) && r<=x(3)/x(11)

rw=G\_temp(2,1);

elseif r>x(3)/x(11) && r<=x(4)/x(11)

rw=G\_temp(3,1);

elseif r>x(4)/x(11) && r<=x(5)/x(11)

rw=G\_temp(4,1);

elseif r>x(5)/x(11) && r<=x(6)/x(11)

rw=G\_temp(5,1);

elseif r>x(6)/x(11) && r<=x(7)/x(11)

rw=G\_temp(6,1);

elseif r>x(7)/x(11) && r<=x(8)/x(11)

rw=G\_temp(7,1);

elseif r>x(8)/x(11) && r<=x(9)/x(11)

rw=G\_temp(8,1);

elseif r>x(9)/x(11) && r<=x(10)/x(11)

rw=G\_temp(9,1);

elseif r>x(10)/x(11) && r<=x(11)/x(11)

rw=G\_temp(10,1);

end

G\_temp(i,:)=[rw];

end

else %tournament selection 抓2

for i=1:10

r=randperm(10,2);

t=[f\_fit(r(1)),f\_fit(r(2))] ;

T=find(t==max(t));

temp(i)=[ G\_temp(r(T(1)),1) ];

end

G\_temp=temp';

end

%crossover(rate=0.8)

G\_cr=zeros(8,1);

for i=1:2:7

while 1

sigma=rand\*2-1; %-1~1

p1=G\_temp(i)+sigma\*(G\_temp(i)+G\_temp(i+1));

p2=G\_temp(i+1)-sigma\*(G\_temp(i)-G\_temp(i+1));

if p1<=10 && p1>=-10 && p2<=10 && p2>=-10

G\_cr(i)=p1;

G\_cr(i+1)=p2;

break;

end

end

end

G\_temp(1:8)=G\_cr(1:8);

%mutation (rate=0.01)

if k~=0 && mod(k,1)==0

while 1

r=randperm(10,1)

noise= rand\*2-1;%-1~1

p=G\_temp(r)+s\*noise;

if p<=10 && p>=-10

G\_temp(r)=p;

break

end

end

end

k=k+1;

end

end

figure,subplot(211),

stem(X(:),Y(:),'filled');

axis([-10 10 0 165]);

grid on

title(['(' num2str(X(find(Y==max(Y)))) ' , ' num2str(max(Y(:))) ')']);

uu=1;

for u=-10:0.1:10

U(uu)=f(u);

uu=uu+1;

end

subplot(212),stem(-10:0.1:10,U);

hold on,grid on

stem(X(find(Y==max(Y))),max(Y(:)),'filled','r');

%%

%EA

clear,clc;

reproduction=2; %1:r.w.select. 2: t.select

q=5;

X=zeros(1,q);

Y=zeros(1,q);

for e=1:q %執行次數

k=0;%世代

r = rand(10,1)\*20-10;% -10~10

G\_temp=[ r ];

while 1

x(1)=0;

for i=1:10

f\_fit(i)=f(G\_temp(i,1))^6;

G(k+1,i)= (G\_temp(i,1));%x

G(k+1,i+10)=f(G\_temp(i,1));%f(x)

G(k+1,i+20)=f\_fit(i);

x(i+1)=x(i)+f\_fit(i);

end

G(k+1,31)=max(G(k+1,21:30));%f\_fit max

G(k+1,32)=x(11)/10;%f\_fit average

if k>100

if G(k+1,31)==G(k,31) && G(k,31)==G(k-1,31) && G(k-1,31)==G(k-2,31)

figure,

Y(e)=max(G(k+1,11:20));

o=find(G(k+1,11:20)==max(G(k+1,11:20)));

X(e)=G(k+1,o(1));

plot(0:k,G(1:k+1,31));

hold on;

plot(0:k,G(1:k+1,32),'--');

title(['(' num2str(X(e)) ' , ' num2str(Y(e)) ')']);

break;

end

end

if reproduction==1

%roulette wheel selection

for i=1:10

r=rand;

if r>=x(1)/x(11) && r<=x(2)/x(11)

rw=G\_temp(1,1);

elseif r>x(2)/x(11) && r<=x(3)/x(11)

rw=G\_temp(2,1);

elseif r>x(3)/x(11) && r<=x(4)/x(11)

rw=G\_temp(3,1);

elseif r>x(4)/x(11) && r<=x(5)/x(11)

rw=G\_temp(4,1);

elseif r>x(5)/x(11) && r<=x(6)/x(11)

rw=G\_temp(5,1);

elseif r>x(6)/x(11) && r<=x(7)/x(11)

rw=G\_temp(6,1);

elseif r>x(7)/x(11) && r<=x(8)/x(11)

rw=G\_temp(7,1);

elseif r>x(8)/x(11) && r<=x(9)/x(11)

rw=G\_temp(8,1);

elseif r>x(9)/x(11) && r<=x(10)/x(11)

rw=G\_temp(9,1);

elseif r>x(10)/x(11) && r<=x(11)/x(11)

rw=G\_temp(10,1);

end

G\_temp(i,:)=[rw];

end

else %tournament selection 抓2

for i=1:10

r=randperm(10,2);

t=[f\_fit(r(1)),f\_fit(r(2))] ;

T=find(t==max(t));

temp(i)=[ G\_temp(r(T(1)),1) ];

end

G\_temp=temp';

end

%crossover(rate=0.8)

G\_cr=zeros(8,1);

for i=1:2:7

while 1

r=rand(1,2); %0~1

p1=G\_temp(i)\*r(1)+G\_temp(i+1)\*(1-r(1));

p2= G\_temp(i)\*r(2)+G\_temp(i+1)\*(1-r(2));

if p1<=10 && p1>=-10 && p2<=10 && p2>=-10

G\_cr(i)=p1;

G\_cr(i+1)=p2;

break;

end

end

end

G\_temp(1:8)=G\_cr(1:8);

%mutation (rate=0.01)

if k~=0 && mod(k,10)==0

while 1

n=randperm(10,1);

d=randperm(10,1);

r=rand;

p=G\_temp(n)+r\*d;

if p<=10 && p>=-10

G\_temp(n)=p;

break

end

end

end

G\_temp(:,1)

k=k+1;

end

end

figure,subplot(211),stem(X(:),Y(:),'filled');

axis([-10 10 0 165]);grid on,title(['(' num2str(X(find(Y==max(Y)))) ' , ' num2str(max(Y(:))) ')']);

uu=1;

for u=-10:0.1:10

U(uu)=f(u);

uu=uu+1;

end

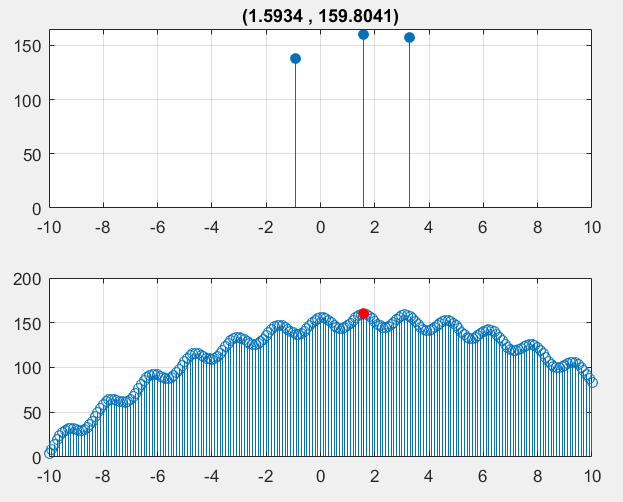
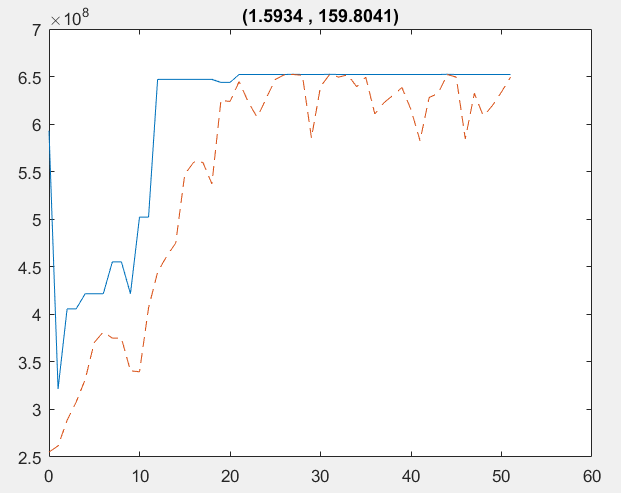
subplot(212),stem(-10:0.1:10,U);

hold on,grid on,stem(X(find(Y==max(Y))),max(Y(:)),'filled','r');

## graph ( c.r.=0.8 , m.r.=0.01 )

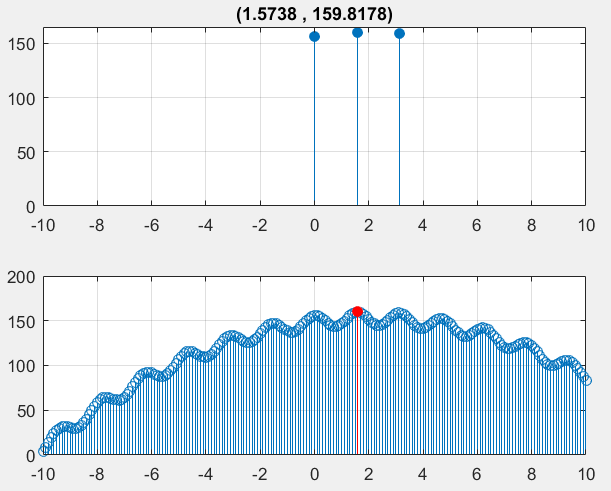
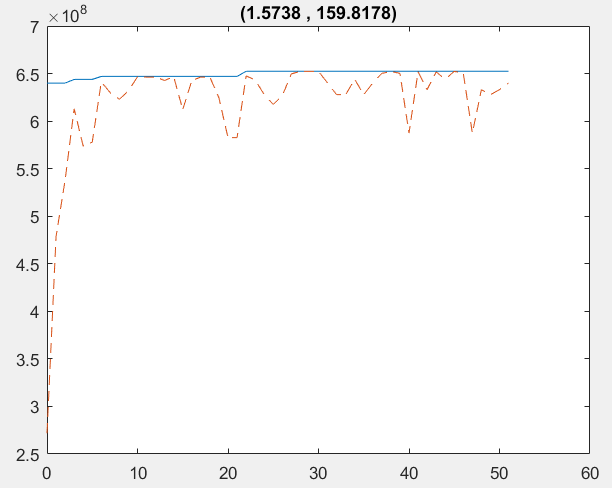
### binary-code G.A. (f\_fit=F(x)^4)

1. Reproduction(輪盤法) & Crossover(Two-point) & Mutation (one-bit)

圖一

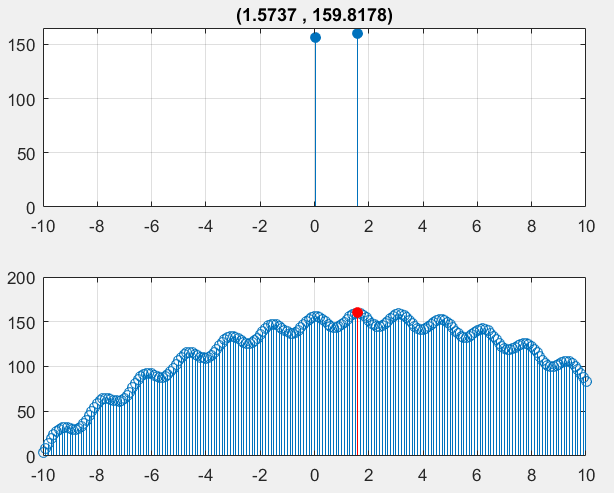
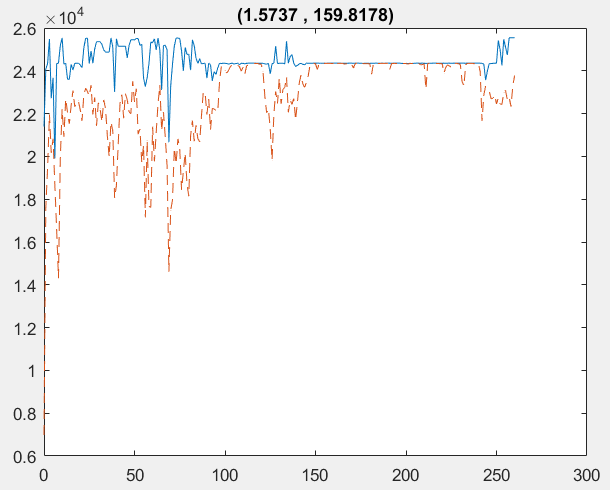
1. Reproduction(競爭法) & Crossover(Two-point) & Mutation (one-bit)

圖二

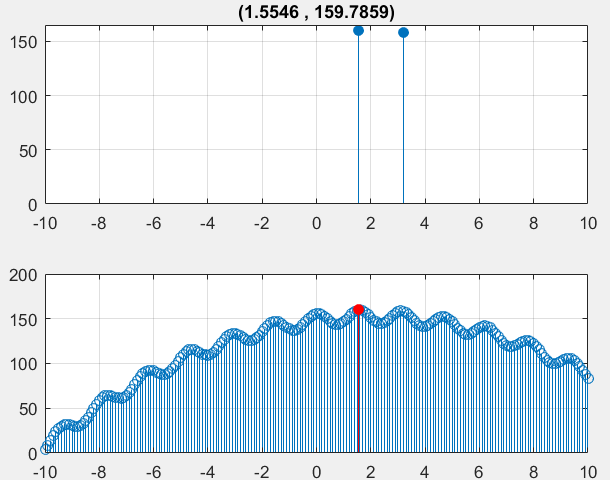
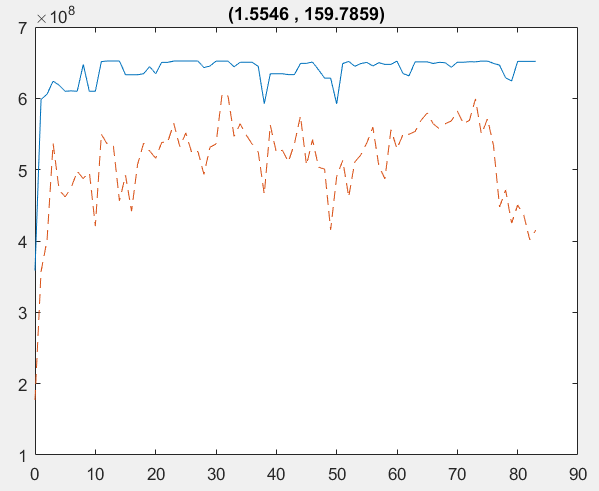
### real-valued G.A. (f\_fit=F(x)^4)

1. Reproduction (輪盤法) & Crossover(randomσ) & Mutation(s=5)

圖三

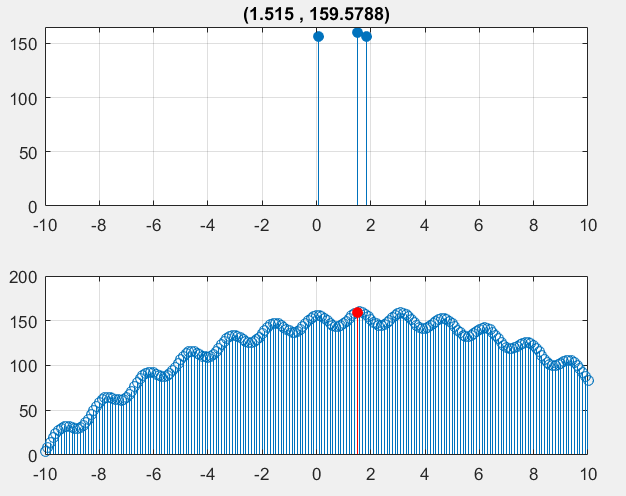
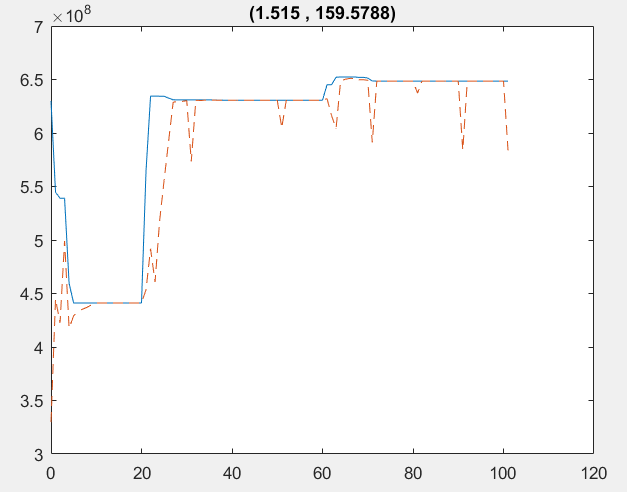
1. Reproduction(競爭法) & Crossover (randomσ)& Mutation(s=5)

圖四

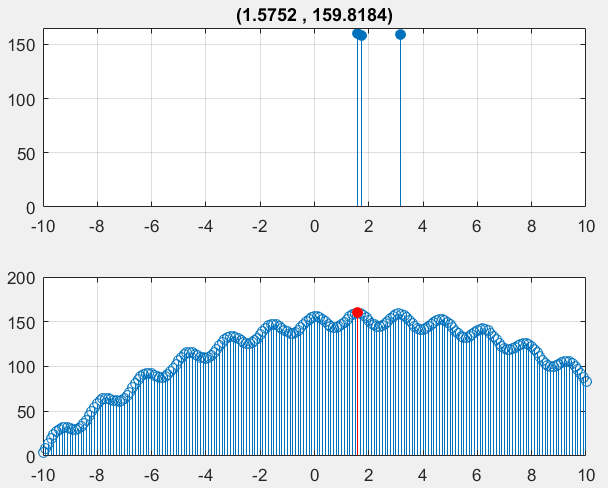
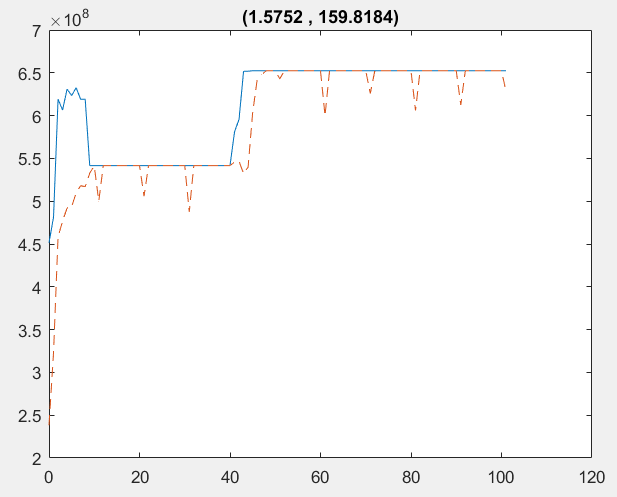
### e.a. (f\_fit=F(x)^4)

1. Reproduction(輪盤法) & Crossover( convex combination ) & Mutation( random d∈ [-20,20] )

圖五

1. Reproduction(競爭法) & Crossover( convex combination ) & Mutation( random d∈ [-20,20] )

圖六

## table

|  |  |  |  |
| --- | --- | --- | --- |
| **複製方式 \ 演化方式** | Binary-code GA | Real-valued GA | E.A. |
| 輪盤法 | 159.8041 | 159.8178 | 159.5788 |
| 競爭法 | 159.8178 | 159.7859 | 159.8184 |

表一. 不同複製方式以及不同演化方式之max(f(x))結果

# conclusion-analysis

在輪盤法中，選出一個好的適應函數是至關重要的，如果選得不好，無法凸顯好基因的優勢，在複製的過程中很可能會把弱勢基因留住，導致最佳解將會卡在局部最大值中，而在競爭法中，由於複製過程只隨機抓取候選基因來比較優劣，適應函數只要能讓我們比較基因之優劣即可，不須特別將好基因凸顯出來，而值得注意的是，競爭法可將最不適應(排名最後)的基因淘汰，而輪盤法則有可能一直選到適應性低之基因。

此次二進制的編碼限制在10個位元，與Real-valued方法相比之下，二進制的精度比較低一些，而Real-valued雖然精度較高，但耗費時間較長，但由於此次模擬數值皆取至小數點後四位，所以10位元的精度(1/1024=0.00097)與Real-valued之精度相差不大，導致模擬之最佳解相近，並沒有凸顯實數解之特性。