**Homework** 2:

<Hw1>

1. Find the decimal value of the binary value [1 0 1 0 1 0 1 0]
2. ------”------real--------------------------”------------------------------ for the range 0 < x <20
3. Create 10 initial populations with 8-bit and 3 genes

<Hw2>

Find the fitness function for

[0 0 0 0 0 0 0 0], [0 0 0 0 1 0 1 0],

[0 0 1 0 1 0 1 0], [1 0 1 0 1 0 1 0],

[1 1 1 1 1 1 1 1] and the corresponding real value in minimizing

<Hw3>

Find the initial population and new population using roulette wheel selection for popu\_size=10; bit\_length=8; in maximizing

**Solution**

<Hw1>



Using Matlab bi2de function:

>> bi2de(fliplr([1 0 1 0 1 0 1 0]))

ans = 170

which could be as wll found as following:

real = 0;

b = [1 0 1 0 1 0 1 0]

for i = 1 : length(b)

real = real + b(i)\*2^(length(b)-i)

end



for the range 0 < x <20:

Matlab code:

function real = GA\_bit2num(bit, range)

real = bi2de(fliplr(bit))/(2^length(bit)-1)\*(range(2)-range(1))+range(1);

>> GA\_bit2num([1 0 1 0 1 0 1 0], [0 20])

ans = 13.3333

To create population I wrote a subroutine similar to the one in the slides:

function initpopu = GA\_initpopu(popu\_size, bit\_length, gene\_number)

initpopu=rand(popu\_size, bit\_length, gene\_number) >= 0.5;

>> GA\_initpopu(10, 8, 3)

|  |  |  |
| --- | --- | --- |
| ans(:,:,1) =  0 1 1 0 1 0 1 0  0 1 0 0 1 1 1 0  0 1 0 0 0 0 1 0  1 1 0 0 0 0 0 0  1 1 0 1 0 0 1 1  0 1 1 0 1 1 0 0  1 0 0 1 1 1 0 0  1 0 1 0 0 0 0 0  1 0 1 1 0 1 1 0  0 0 1 1 0 0 0 0 | ans(:,:,2) =  1 0 0 0 1 1 1 0  0 1 1 0 1 0 1 0  0 1 1 1 1 0 1 1  0 1 1 0 0 1 0 0  0 0 1 1 0 0 0 0  0 1 1 1 0 1 0 1  1 0 1 0 1 1 1 0  1 0 0 0 1 1 1 1  1 1 1 1 1 1 1 0  0 0 1 0 0 0 0 0 | ans(:,:,3) =  1 0 0 0 0 0 0 0  0 1 0 1 1 0 0 0  0 1 1 0 1 0 1 0  0 0 0 0 1 0 1 1  1 1 1 1 0 0 1 0  0 1 0 1 1 0 0 1  0 0 0 0 0 1 0 0  0 1 0 0 0 1 1 1  0 0 1 0 0 1 1 0  1 1 0 0 0 0 0 0 |

**<Hw2>**

Fitness function is:

function PI = GA\_fitfun(chro)

global MIN\_offset

MIN\_offset = 100;

x = chro(1);

y = chro(2);

z = x\*cos(x)+y\*cos(y);

PI = MIN\_offset - z;

To find corresponding x, y values GA\_bit2num can be used, to find f(x,y) value the following function can be used:

function Z = GA\_funz(popu\_real)

for i=1:size(popu\_real,1)

x = popu\_real(i,1);

y = popu\_real(i,2);

Z(i) = x\*cos(x)+y\*cos(y);

end

**Result:**

[fitness, popu\_real, popu] = GA\_fitpopu([0 0 0 0 0 0 0 0; 0 0 0 0 1 0 1 0; 0 0 1 0 1 0 1 0; ...

1 0 1 0 1 0 1 0; 1 1 1 1 1 1 1 1], 4, [0 0; 10 10], 'GA\_fitfun')

Z = GA\_funz(popu\_real)

Output:

fitness =

116.7814 100.0000 93.8175 93.5039 87.6351

popu\_real =

10.0000 10.0000

0 0

0 6.6667

1.3333 6.6667

6.6667 6.6667

popu =

1 1 1 1 1 1 1 1

0 0 0 0 0 0 0 0

0 0 0 0 1 0 1 0

0 0 1 0 1 0 1 0

1 0 1 0 1 0 1 0

Z =

-16.7814 0 6.1825 6.4961 12.3649

**<Hw3>**

**Execute:**

[fitness, popu\_real, popu] = GA\_fitpopu(GA\_initpopu(10,8,1),8,[0;10],'GA\_fitfunXXX')

new\_popu = GA\_wheel(fitness, popu)

for i=1:size(new\_popu, 1)

new\_popu\_real(i) = GA\_bit2num(new\_popu(i,:), [0;10]);

end

**Output**:

|  |  |
| --- | --- |
| popu =  1 1 1 1 0 0 0 0  1 1 1 0 0 1 1 1  1 1 0 0 1 1 1 0  1 1 0 0 1 0 0 1  1 1 0 0 0 0 1 1  1 1 0 0 0 0 0 0  1 0 0 1 1 1 1 1  0 1 1 1 1 1 0 1  0 0 1 0 1 1 1 0  0 0 0 0 1 0 1 1 | new\_popu =  1 1 1 1 0 0 0 0  1 1 1 1 0 0 0 0  1 1 1 0 0 1 1 1  1 1 0 0 1 1 1 0  1 1 0 0 1 0 0 1  1 1 0 0 0 0 1 1  1 1 0 0 0 0 0 0  1 1 1 1 0 0 0 0  1 1 1 1 0 0 0 0  1 1 1 0 0 1 1 1 |
| popu\_real =  9.4118  9.0588  8.0784  7.8824  7.6471  7.5294  6.2353  4.9020  1.8039  0.4314 | new\_popu\_real =  9.4118  9.4118  9.0588  8.0784  7.8824  7.6471  7.5294  9.4118  9.4118  9.0588 |