**“ GENETIC ALGORITHM ”**

**EXERCISE:**

**Object 01:** Implement genetic algorithm on a population of n strings. Suppose that l=10,n=6 we toss a coin 60 times & get the population .Calculate the fitness function 3 times and print which fitness function is better.

**Source Code:**

odd=[[1,1,1,1,0,1,0,1,0,1],[0,1,1,1,0,0,0,1,0,1],[1,1,1,0,1,1,0,1,0,1],[0,1,0,0,0,1,0,0,1,1],[1,1,1,0,1,1,1,1,0,1],[0,1,0,0,1,1,0,0,0,0]]

arrange=[]

even = []

newadd=[]

crossover=2

add=0

print('Initialization')

for i in range(6):

b=0

for j in range(10):

if(odd[i][j]==1):

b=b+1

print(odd[i][j],end= ' ')

print('initial fitnes =',b)

primis=arrange

primis.append(b)

add=add+b

print('Total Fitness= '+str(add))

primis.sort(reverse=True)

print('Arrange in Assending order')

for i in range(6):

for j in range(10):

if primis[0]==8:

even.append(odd[4])

if primis[1]==7:

even.append(odd[0])

if primis[2]==7:

even.append(odd[2])

if primis[3]==5:

even.append(odd[1])

if primis[4]==4:

even.append(odd[3])

if primis[5]==3:

even.append(odd[5])

for k in range(6):

print(str(even[k])+'='+str(primis[k]))

print('Crossover after '+str(crossover)+' bits')

even[0],even[1],even[4],even[5]=even[1],even[0],even[5],even[4]

for i in range(6):

for j in range(crossover,10):

print(even[i][j],end= ' ')

print('\n')

print('Over all result after crossover')

for i in range(6):

for j in range(10):

print(even[i][j],end= ' ')

print('\n')

for k in range(3):

print('mutation at '+str((k+1))+' process')

for i in range(6):

for j in range(10):

if(even[i][j]==0)&(k==0):

even[2][3]=1

even[4][6]=1

even[5][3]=1

elif(even[i][j]==0)&(k==1):

even[2][3],even[2][8]=1,1

even[4][6]=1

even[5][3]=1

elif(even[i][j]==0)&(k==2):

even[2][3]=1

even[4][6]=1

even[5][0],even[5][2],even[5][3]=1,1,1

add=0

for i in range(6):

b2=0

for j in range(10):

if(even[i][j]==1):

b2=b2+1

print(even[i][j],end= ' ')

print('final fitnes =',b2)

add=add+b2

print('total fitness= '+str(add))

newadd.append(add)

print('Comparison between the final total fitnesses',newadd)

if(newadd[0]>newadd[1])&(newadd[0]>newadd[2]):

print(newadd[0]+" is better then other fitness")

elif(newadd[1]>newadd[0])&(newadd[1]>newadd[2]):

print(newadd[1]+" is better then other fitness")

elif(newadd[2]>newadd[1])&(newadd[2]>newadd[0]):

print(str(newadd[2])+" is better then other fitness")

**Output:**

Initialization

1 1 1 1 0 1 0 1 0 1 initial fitnes = 7

0 1 1 1 0 0 0 1 0 1 initial fitnes = 5

1 1 1 0 1 1 0 1 0 1 initial fitnes = 7

0 1 0 0 0 1 0 0 1 1 initial fitnes = 4

1 1 1 0 1 1 1 1 0 1 initial fitnes = 8

0 1 0 0 1 1 0 0 0 0 initial fitnes = 3

Total Fitness= 34

Arrange in Assending order

[1, 1, 1, 0, 1, 1, 1, 1, 0, 1]=8

[1, 1, 1, 1, 0, 1, 0, 1, 0, 1]=7

[1, 1, 1, 0, 1, 1, 0, 1, 0, 1]=7

[0, 1, 1, 1, 0, 0, 0, 1, 0, 1]=5

[0, 1, 0, 0, 0, 1, 0, 0, 1, 1]=4

[0, 1, 0, 0, 1, 1, 0, 0, 0, 0]=3

Crossover after 2 bits

1 1 0 1 0 1 0 1

1 0 1 1 1 1 0 1

1 0 1 1 0 1 0 1

1 1 0 0 0 1 0 1

0 0 1 1 0 0 0 0

0 0 0 1 0 0 1 1

Over all result after crossover

1 1 1 1 0 1 0 1 0 1

1 1 1 0 1 1 1 1 0 1

1 1 1 0 1 1 0 1 0 1

0 1 1 1 0 0 0 1 0 1

0 1 0 0 1 1 0 0 0 0

0 1 0 0 0 1 0 0 1 1

mutation at 1 process

1 1 1 1 0 1 0 1 0 1 final fitnes = 7

1 1 1 0 1 1 1 1 0 1 final fitnes = 8

1 1 1 1 1 1 0 1 0 1 final fitnes = 8

0 1 1 1 0 0 0 1 0 1 final fitnes = 5

0 1 0 0 1 1 1 0 0 0 final fitnes = 4

0 1 0 1 0 1 0 0 1 1 final fitnes = 5

total fitness= 37

mutation at 2 process

1 1 1 1 0 1 0 1 0 1 final fitnes = 7

1 1 1 0 1 1 1 1 0 1 final fitnes = 8

1 1 1 1 1 1 0 1 1 1 final fitnes = 9

0 1 1 1 0 0 0 1 0 1 final fitnes = 5

0 1 0 0 1 1 1 0 0 0 final fitnes = 4

0 1 0 1 0 1 0 0 1 1 final fitnes = 5

total fitness= 38

mutation at 3 process

1 1 1 1 0 1 0 1 0 1 final fitnes = 7

1 1 1 0 1 1 1 1 0 1 final fitnes = 8

1 1 1 1 1 1 0 1 1 1 final fitnes = 9

0 1 1 1 0 0 0 1 0 1 final fitnes = 5

0 1 0 0 1 1 1 0 0 0 final fitnes = 4

1 1 1 1 0 1 0 0 1 1 final fitnes = 7

total fitness= 40

Comparison between the final total fitnesses [37, 38, 40]

40 is better then other fitness

**Object 02:**Consider the above scenario utilize the population by asking the user & mutation the bits randomly .calculate the fitness function 3 times and print which fitness function is better

**Source Code:**

import random

odd = [[0 for col in range(10)] for row in range(6)]

for row in range(6):

for col in range(10):

c=int(input())

odd[row][col]= c

print('Input values')

print(odd)

arrange=[]

even = []

newadd=[]

crossover=2

add=0

print('Initialization')

for i in range(6):

b=0

for j in range(10):

if(odd[i][j]==1):

b=b+1

print(odd[i][j],end= ' ')

print('initial fitnes =',b)

primis=arrange

primis.append(b)

add=add+b

print('Total Fitness= '+str(add))

primis.sort(reverse=True)

print('Arrange in Assending order')

for i in range(6):

for j in range(10):

if primis[0]==8:

even.append(odd[4])

if primis[1]==7:

even.append(odd[0])

if primis[2]==7:

even.append(odd[2])

if primis[3]==5:

even.append(odd[1])

if primis[4]==4:

even.append(odd[3])

if primis[5]==3:

even.append(odd[5])

for k in range(6):

print(str(even[k])+'='+str(primis[k]))

print('Crossover after '+str(crossover)+' bits')

even[0],even[1],even[4],even[5]=even[1],even[0],even[5],even[4]

for i in range(6):

for j in range(crossover,10):

print(even[i][j],end= ' ')

print('\n')

print('Over all result after crossover')

for i in range(6):

for j in range(10):

print(even[i][j],end= ' ')

print('\n')

for k in range(3):

print('mutation at '+str((k+1))+' process')

for i in range(6):

value1=random.randint(0,5)

for j in range(10):

value2=random.randint(0,9)

if(even[i][j]==0)&(k==0):

even[value1][value2]=1

if(even[i][j]==0)&(k==1):

even[value1][value2]=1

if(even[i][j]==0)&(k==2):

even[value1][value2]=1

add=0

for i in range(6):

b2=0

for j in range(10):

if(even[i][j]==1):

b2=b2+1

print(even[i][j],end= ' ')

print('final fitnes =',b2)

add=add+b2

print('total fitness= '+str(add))

newadd.append(add)

print('Comparison between the final total fitnesses',newadd)

if(newadd[0]>newadd[1])&(newadd[0]>newadd[2]):

print(newadd[0]+" is better then other fitness")

elif(newadd[1]>newadd[0])&(newadd[1]>newadd[2]):

print(newadd[1]+" is better then other fitness")

elif(newadd[2]>newadd[1])&(newadd[2]>newadd[0]):

print(str(newadd[2])+" is better then other fitness")

**Output:**

Input values

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

Initialization

1 1 0 0 0 1 1 1 1 0 initial fitnes = 6

1 0 1 0 1 0 1 0 1 1 initial fitnes = 6

1 0 1 0 0 0 0 1 0 1 initial fitnes = 4

0 1 0 0 1 0 1 0 1 1 initial fitnes = 5

0 1 0 1 0 1 1 0 1 0 initial fitnes = 5

1 0 1 0 1 0 0 0 0 0 initial fitnes = 3

Total Fitness= 29

Arrange in Assending order

[1, 0, 1, 0, 1, 0, 1, 0, 1, 1]=6

[0, 1, 0, 0, 1, 0, 1, 0, 1, 1]=6

[1, 0, 1, 0, 1, 0, 0, 0, 0, 0]=5

[1, 0, 1, 0, 1, 0, 1, 0, 1, 1]=5

[0, 1, 0, 0, 1, 0, 1, 0, 1, 1]=4

[1, 0, 1, 0, 1, 0, 0, 0, 0, 0]=3

Crossover after 2 bits

0 0 1 0 1 0 1 1

1 0 1 0 1 0 1 1

1 0 1 0 0 0 0 0

1 0 1 0 1 0 1 1

1 0 1 0 0 0 0 0

0 0 1 0 1 0 1 1

Over all result after crossover

0 1 0 0 1 0 1 0 1 1

1 0 1 0 1 0 1 0 1 1

1 0 1 0 1 0 0 0 0 0

1 0 1 0 1 0 1 0 1 1

1 0 1 0 1 0 0 0 0 0

0 1 0 0 1 0 1 0 1 1

mutation at 1 process

1 1 1 1 1 0 1 0 1 1 final fitnes = 8

1 1 1 0 1 1 1 0 1 1 final fitnes = 8

1 0 1 1 1 0 1 0 1 1 final fitnes = 7

1 1 1 0 1 1 1 0 1 1 final fitnes = 8

1 0 1 1 1 0 0 0 1 1 final fitnes = 6

1 1 1 1 1 0 1 0 1 1 final fitnes = 8

total fitness= 45

mutation at 2 process

1 1 1 1 1 0 1 1 1 1 final fitnes = 9

1 1 1 0 1 1 1 0 1 1 final fitnes = 8

1 0 1 1 1 1 1 1 1 1 final fitnes = 9

1 1 1 0 1 1 1 0 1 1 final fitnes = 8

1 0 1 1 1 1 1 1 1 1 final fitnes = 9

1 1 1 1 1 0 1 1 1 1 final fitnes = 9

total fitness= 52

mutation at 3 process

1 1 1 1 1 1 1 1 1 1 final fitnes = 10

1 1 1 1 1 1 1 0 1 1 final fitnes = 9

1 0 1 1 1 1 1 1 0 1 final fitnes = 8

1 1 1 1 1 1 1 0 1 1 final fitnes = 9

1 0 1 1 1 1 1 0 0 1 final fitnes = 7

1 1 1 1 1 1 1 1 1 1 final fitnes = 10

total fitness= 56

Comparison between the final total fitnesses [45, 52, 53]

56 is better then other fitness