REPORT

Coding assignment 2

Genetic Algorithm

SUBMITTED BY

ADITYA GANGAN

204103306

- **Aim**: To minimize the function

$$f\left(x_1, x_2\right) = x_1 + x_2 - 2x_1^{\,2} - x_2^{\,2} + x_1 x_2$$

using binary coded Genetic algorithm.

- **Given range of input variables**:

$$0.0 \le x_1 \le 0.5$$

$$0.0 \le x_2 \le 0.5$$

- **Features of the algorithm**:

Reproduction scheme used	: Roulette wheel selection
Type of crossover	: Two point crossover
Mutation	: Bitwise

- **Inputs taken from the user**:

1) Objective function
2) Population Size
3) Crossover probability
4) Mutation probability

A code using genetic algorithm was made and the given problem was solved using the following **input parameters**;

1) Fitness function used : $-f(x)$
2) Population size : 50
3) Crossover probability : 0.98
4) Mutation probability : 0.01
5) Maximum number of iterations performed : 300
6) Length of string for each variable : 20

- **Results:**

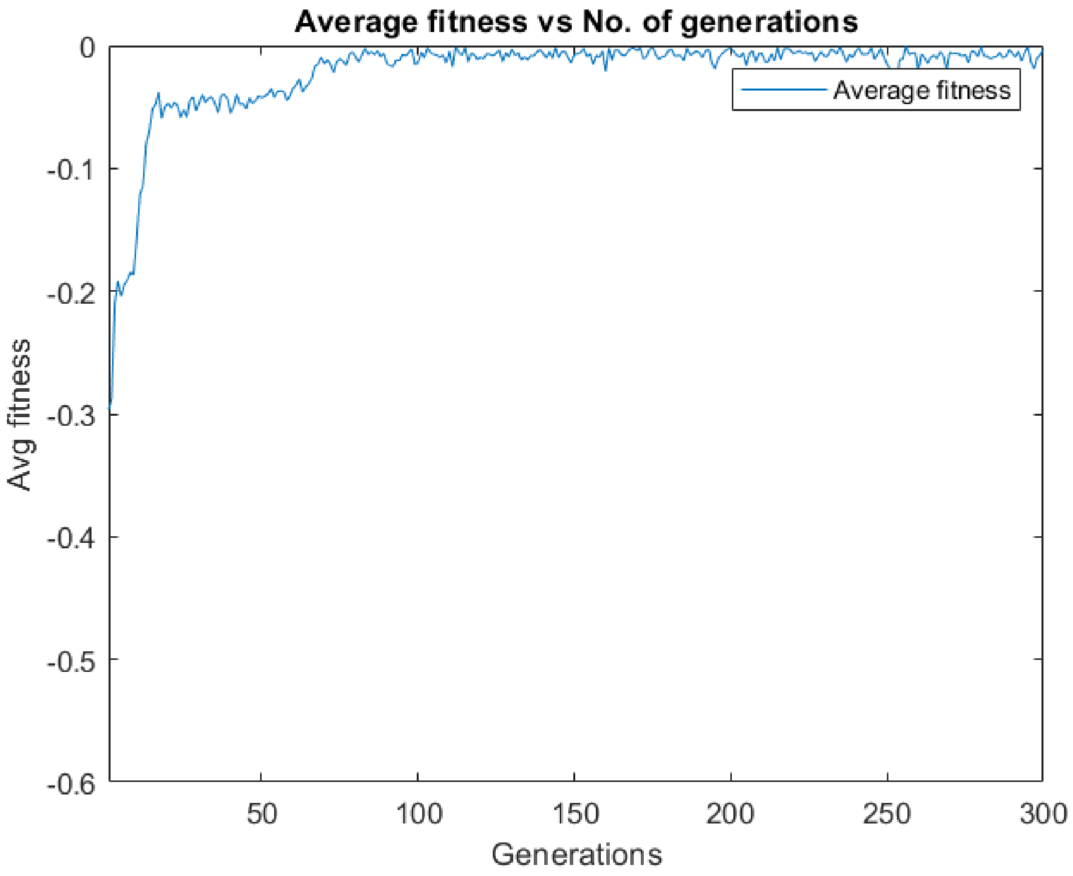
When the code was run for these parameters, the following output was obtained;

Solution no.	x_1	x_2	Fitness
1	6.10352144577164e-05	4.76837612950910e-07	-6.15046303523714e-05
2	0	0.0625005364423146	-0.0585942193867375
3	6.10352144577164e-05	0	-6.10277638629086e-05
4	0	4.76837612950910e-07	-4.76837385576800e-07
5	0.0312500298023508	4.76837612950910e-07	-0.0292973928156304
6	0.000550270605345350	4.76837612950910e-07	-0.000550142109642434
7	0	0.0312505066399638	-0.0302739124747093
8	0	0.000122547266528384	-0.000122532248695850
9	0	4.76837612950910e-07	-4.76837385576800e-07
10	0.0234375223517631	1.43051283885273e-06	-0.0223403514822534
11	0	2.38418806475455e-06	-2.38418238040182e-06
12	0	4.76837612950910e-07	-4.76837385576800e-07
13	0	4.76837612950910e-07	-4.76837385576800e-07
14	0	4.76837612950910e-07	-4.76837385576800e-07
15	0.00781250745058770	4.76837612950910e-07	-0.00769091746793971
16	0	4.76837612950910e-07	-4.76837385576800e-07
17	0	4.76837612950910e-07	-4.76837385576800e-07
18	0.000244140857830866	4.76837612950910e-07	-0.000244498602115062
19	0	4.76837612950910e-07	-4.76837385576800e-07
20	0	4.76837612950910e-07	-4.76837385576800e-07
21	0	3.09944448418091e-05	-3.09934841861981e-05
22	0	4.76837612950910e-07	-4.76837385576800e-07
23	0	4.76837612950910e-07	-4.76837385576800e-07
24	0	4.76837612950910e-07	-4.76837385576800e-07

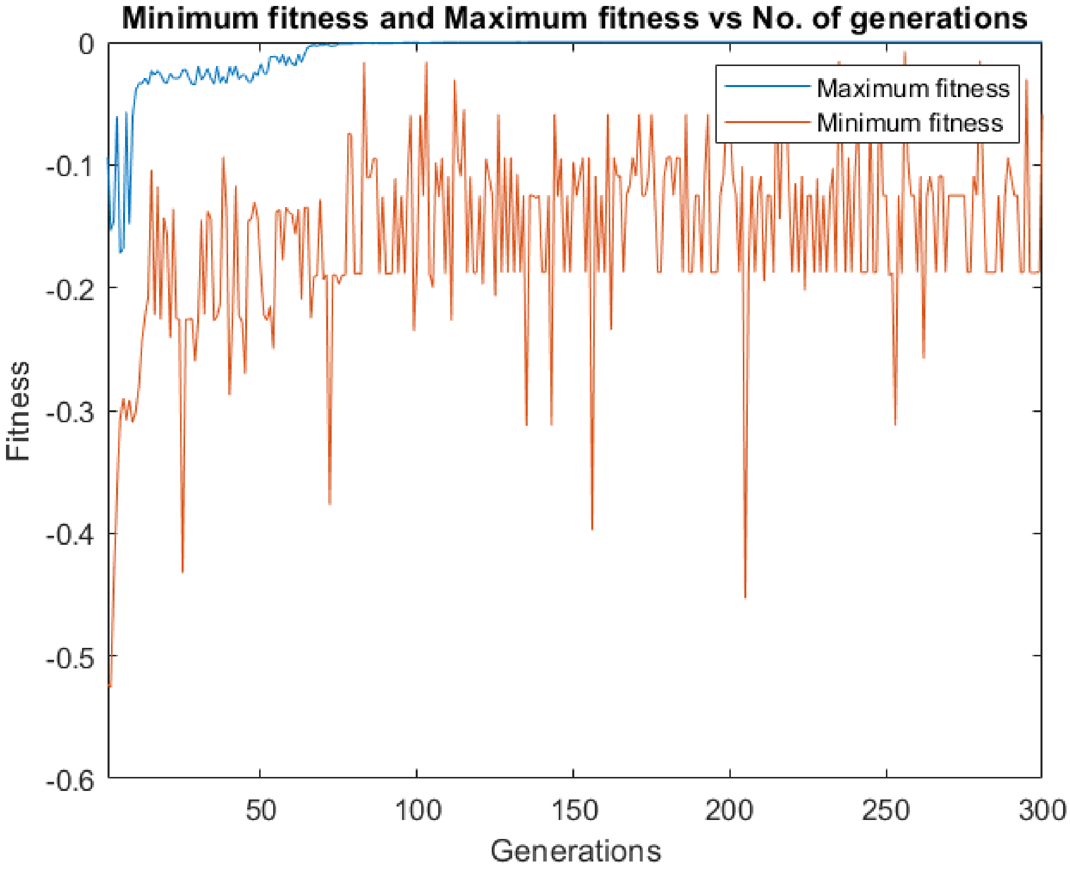
25	0	4.76837612950910e-07	-4.76837385576800e-07
26	0	4.76837612950910e-07	-4.76837385576800e-07
27	0	4.76837612950910e-07	-4.76837385576800e-07
28	0	4.76837612950910e-07	-4.76837385576800e-07
29	0.00196838566626135	4.76837612950910e-07	-0.00196111435798497
30	0	0.000488758553274682	-0.000488519668351283
31	0	4.76837612950910e-07	-4.76837385576800e-07
32	0	4.76837612950910e-07	-4.76837385576800e-07
33	0.000244140857830866	4.76837612950910e-07	-0.000244498602115062
34	7.62940180721455e-06	4.76837612950910e-07	-8.10612641523323e-06
35	3.81470090360728e-06	4.76837612950910e-07	-4.29151100429098e-06
36	0	4.76837612950910e-07	-4.76837385576800e-07
37	0	4.76837612950910e-07	-4.76837385576800e-07
38	0	6.15120520706673e-05	-6.15082683381174e-05
39	0.000976563431323463	4.76837612950910e-07	-0.000975133382100419
40	0.000244140857830866	4.76837612950910e-07	-0.000244498602115062
41	0	4.76837612950910e-07	-4.76837385576800e-07
42	0	4.76837612950910e-07	-4.76837385576800e-07
43	0	6.15120520706673e-05	-6.15082683381174e-05
44	0	4.76837612950910e-07	-4.76837385576800e-07
45	0	4.76837612950910e-07	-4.76837385576800e-07
46	0	4.76837612950910e-07	-4.76837385576800e-07
47	0	0.000488758553274682	-0.000488519668351283
48	0	0	0
49	0	4.76837612950910e-07	-4.76837385576800e-07
50	0	4.76837612950910e-07	-4.76837385576800e-07

The following plots were obtained:

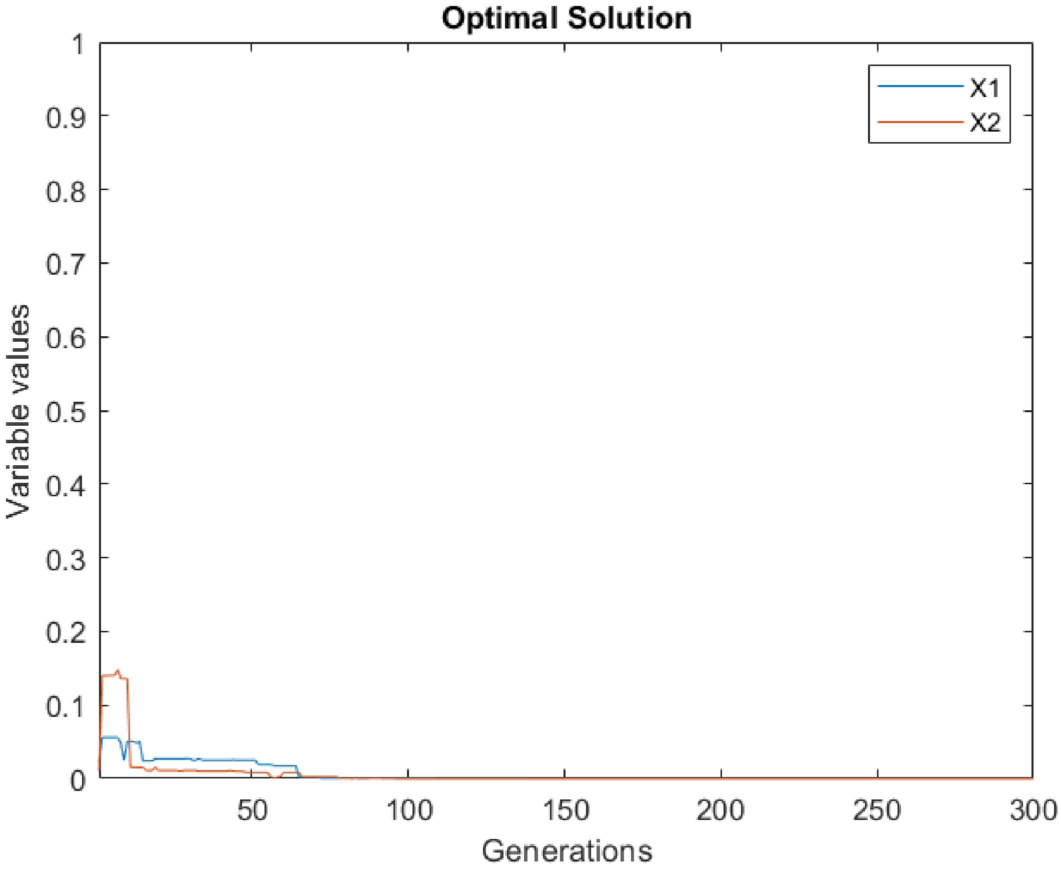
Plot 1:



Plot 2:


Minimum fitness and Maximum fitness vs No. of generations
Fitness
Generations
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
50
100
150
200
250
300
Maximum fitness
Minimum fitness

Plot 3:



Minimum function value = -mode(fitness function value)

= 4.7684e-07

Value of x_1 = 0

Value of x_2 = 4.7684e-07

- **Conclusion**

i) The average fitness tends to converge to 0, as seen in plot 1. However, it never becomes absolutely 0. This can be owed to the fact that genetic algorithm still gives a few random solutions (and corresponding fitness values) apart from the optimum solution, even in the last iteration.

ii) The minimum fitness graph, which never really converges, proves that there will always be a few bad solutions in each iteration.

iii) This randomness arises due to crossover and mutation, which actually helps the algorithm to overcome the problem of getting stuck in local minima.