**Experiment No: 11**

**Aim: Write a program to maximize f(x1+x2)=4x1+3x2 using Genetic Algorithm.**

**Program code:**

import random

POPULATION\_SIZE = 10

MAX\_GENERATIONS = 7

CROSSOVER\_PROBABILITY = 0.7

MUTATION\_PROBABILITY = 0.1

def objective\_function(x1, x2):

return 4 \* x1 + 3 \* x2

def initialize\_population():

print("Initial Population")

population = []

for i in range(POPULATION\_SIZE):

xl = random.uniform(0, 2)

x2 = random.uniform(0, 2)

population.append((xl, x2))

print(f"{i} xl: {xl} x2: {x2}")

return population

def crossover(xl1, xl2, x21, x22):

a = random.random()

new\_xl = a \* xl1 + (1 - a) \* xl2

new\_x2 = a \* x21 + (1 - a) \* x22

return new\_xl, new\_x2

def mutation(x):

if random.random() <= MUTATION\_PROBABILITY:

delta = random.uniform(-1, 1)

x += delta

return x

def main():

population = initialize\_population()

max\_val = 0

max\_xl, max\_x2 = 0, 0

for generation in range(MAX\_GENERATIONS):

print(f"\nAfter {generation} generation")

# Evaluation

for i, (xl, x2) in enumerate(population):

sum\_val = objective\_function(xl, x2)

print(f"{i} xl: {xl} x2: {x2} sum: {sum\_val}")

if sum\_val > max\_val:

max\_val = sum\_val

max\_xl, max\_x2 = xl, x2

# Crossover

for i in range(0, POPULATION\_SIZE, 2):

if random.random() <= CROSSOVER\_PROBABILITY:

xl1, x21 = population[i]

xl2, x22 = population[i + 1]

new\_xl, new\_x2 = crossover(xl1, xl2, x21, x22)

population[i] = (new\_xl, new\_x2)

population[i + 1] = (new\_xl, new\_x2)

# Mutation

for i in range(POPULATION\_SIZE):

xl, x2 = population[i]

xl = mutation(xl)

x2 = mutation(x2)

population[i] = (xl, x2)

# Output

print("\nTHE SOLUTION OF THE FOLLOWING PROBLEM IS")

print("SUM: MAXIMIZE F(xl, x2) = 4xl + 3x2")

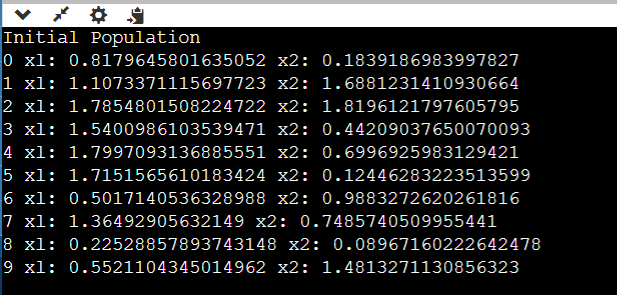
print("2\*x1 + 3\*x2 <= 6\n-3\*x1 + 2\*x2 <= 3\n2\*x1 + x2 <= 4")

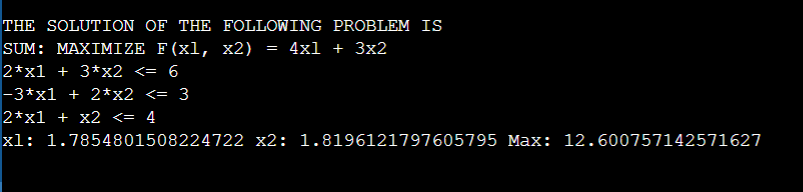
print(f"xl: {max\_xl} x2: {max\_x2} Max: {max\_val}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Output:**

****

****