Name: Jigar Siddhpura **SAPID: 60004200155**

DIV: C/C2 **Branch: Computer Engineering**

AI EXPERIMENT 5 - Genetic Algorithm

60004210155
Jigar Siddhpura

AI - Experiment 5

Aim : Program on genetic algorithm to solve optimization problem.

Theory : 1. Genetic Algorithm (GA's) are heuristic search & optimization techniques inspired by process of natural evolution.

2. They are used to find approximate solutions to optimization & search problems, relying on principles such as selection, crossover, mutation to evolve a population of candidate solutions over several generations.

3. Basic concepts of GA's

a) Initialization - Start by creating an initial population of potential solutions.

b) Fitness Evaluation - Evaluate the fitness of each individual in population. It represents how well a solution solves a problem in hand.

c) Selection - Select individuals based on fitness scores, mimicking the principle of 'survival of fittest'.

d) Crossover - Perform crossover operation to combine genetic material of 2 parents to create one or more offspring.

e) Mutation - Apply mutation with certain probability. It introduces small random changes in offspring, genetic material, promoting genetic diversity.

Advantages of GA's :

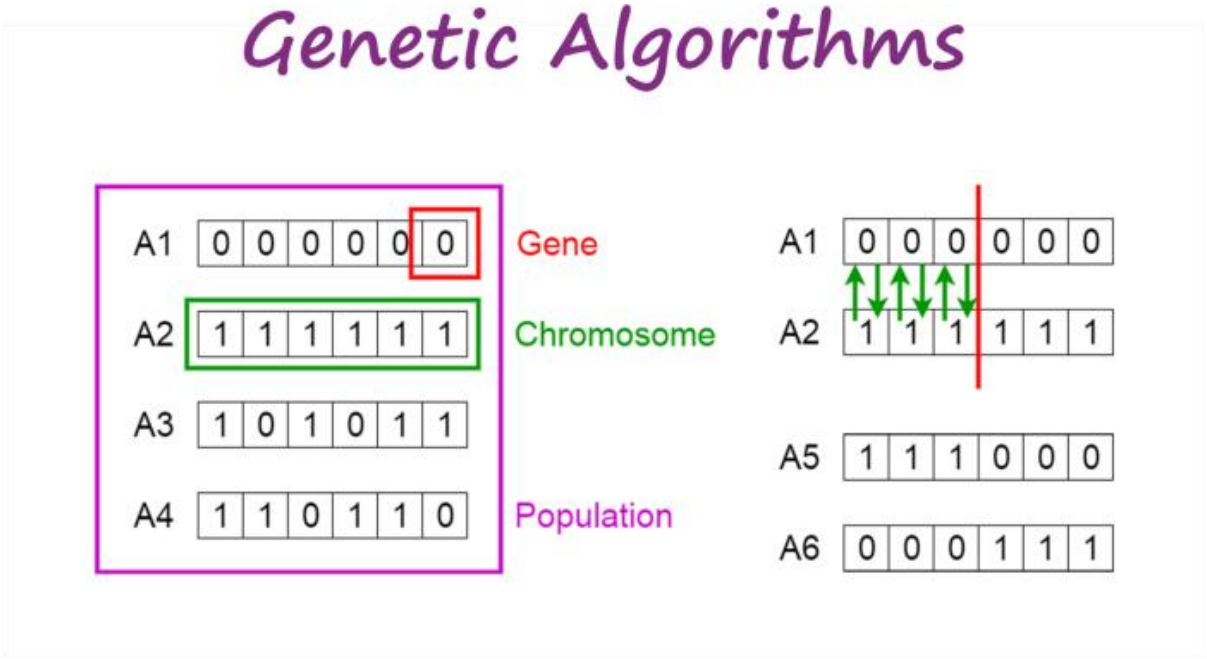
1. They excel in exploration & exploitation, navigating sol^{n} spaces adaptly.
2. Their inherent parallelism enables efficient parallel processing applying operations independently.
3. Ability to tackle diverse problems.

~~Didd~~ Disadvantages of GA's :

1. Computationally intensive
2. Requires proper parameter tuning
3. Might converge prematurely to suboptimal solutions.

Conclusions :

GA's are powerful & versatile optimized techniques inspired by principles of natural evolution. They offer effective approach for solving complex, high dimensional problems where traditional methods might fail.



Code :

```
import random

gene = ['000000','111111','101011','110110']

def selection(gene):
    x = [int(x,2) for x in gene]
    fx = [int(x)*int(x) for x in x]
    fx_sum = sum(fx)
    fx_avg = fx_sum//len(fx)
    excepted_count = [round((i/fx_avg),4) for i in fx]
    actual_count = [round(i) for i in excepted_count]
    mate_pool = []
    for i,j in zip(actual_count,gene):
        if i:
            for c in range(i):
                mate_pool.append(j)
    return x,fx, fx_sum,fx_avg,excepted_count,actual_count,mate_pool

def generate_mate(size,mate_element_size):
    if size % 2 != 0:
        return -1

    available_positions = list(range(size))
    random.shuffle(available_positions)

    mate = [-1] * size
    crossover =[-1] * size

    for i in range(size):
        if mate[i] == -1:
            j = random.choice(available_positions)
            while mate[j] != -1 or j == i:
                j = random.choice(available_positions)
            mate[i] = j
```

```
            mate[j] = i
            available_positions.remove(i)
            available_positions.remove(j)
    for i in mate:
        if crossover.count(-1) != 0:
            crossover[i] = crossover[mate[i]] = random.randint(1,mate_element_size-1)
    return mate,crossover

def crossover(mate_pool):
    mate, crossover_points = generate_mate(len(mate_pool),len(mate_pool[0]))
    new_poplu = [-1]*len(mate_pool)
    for i in mate:
        new_poplu[i] = mate_pool[i][:crossover_points[i]] + mate_pool[mate[i]][crossover_points[i]:]

    x = [int(x,2) for x in new_poplu]
    fx = [int(x)*int(x) for x in x]


    return mate_pool,new_poplu,mate,crossover_points,x, fx


def GA(gene,iter,n):
    if iter == 0:
        return

    x,fx, fx_sum,fx_avg,excepted_count,actual_count,mate_pool = selection(gene)
    if sum(actual_count)!=len(gene):
        print("Error dont know what to do at this situation ")

        return
    print(f"\n------------------------------------------------ GENERATION {n} ----------------------
----------------------------")
    print("Initial Population\tX Value\t\tFitness Value( f(x) )\tProbability(Expected Count)\tActual
Count")
    print(f"-------------------------------------------------------------------------------------------
-------------------------")
    for i in range(len(gene)):
        print(f"{gene[i]}\t\t\t{x[i]}\t\t{fx[i]}\t\t\t{excepted_count[i]}\t\t\t\t{actual_count[i]}")
    mate_pool,new_poplu,mate,crossover_points,x, fx = crossover(mate_pool)
    print(f"\n------------------------------------------------ New Population {n} ------------------
----------------------------")
    print("Mate Pool\tMate\t\tCrossover Points\tNew Population\t\tx value\t\tf(x)")
    print(f"-------------------------------------------------------------------------------------------
-------------------------")
    for i in range(len(gene)):
        print(f"{mate_pool[i]}\t\t{mate[i]}\t\t{crossover_points[i]}\t\t\t{new_poplu[i]}\t\t\t{x[i]}\
t\t{fx[i]}")

    GA(new_poplu,iter-1,n+1)

GA(gene,4,0)
```

Output :

```
------------------------------------------ GENERATION 0 ------------------------------------------
Initial Population      X Value         Fitness Value( f(x) )   Probability(Expected Count)     Actual Count
--------------------------------------------------------------------------------------------------
000000                  0               0                       0.0                             0
111111                  63              3969                    1.8181                          2
101011                  43              1849                    0.847                           1
110110                  54              2916                    1.3358                          1

------------------------------------------ New Population 0 ------------------------------------------
Mate Pool       Mate            Crossover Points        New Population          x value        f(x)
------------------------------------------------------------------------------------------------------
111111          3               5                       111110                  62              3844
111111          2               5                       111111                  63              3969
101011          1               5                       101011                  43              1849
110110          0               5                       110111                  55              3025

------------------------------------------ GENERATION 1 ------------------------------------------
Initial Population      X Value         Fitness Value( f(x) )   Probability(Expected Count)     Actual Count
--------------------------------------------------------------------------------------------------
111110                  62              3844                    1.2122                          1
111111                  63              3969                    1.2517                          1
101011                  43              1849                    0.5831                          1
110111                  55              3025                    0.954                           1

------------------------------------------ New Population 1 ------------------------------------------
Mate Pool       Mate            Crossover Points        New Population          x value        f(x)
------------------------------------------------------------------------------------------------------
111110          3               1                       110111                  55              3025
111111          2               4                       111111                  63              3969
101011          1               4                       101011                  43              1849
110111          0               1                       111110                  62              3844

------------------------------------------ GENERATION 2 ------------------------------------------
Initial Population      X Value         Fitness Value( f(x) )   Probability(Expected Count)     Actual Count
--------------------------------------------------------------------------------------------------
110111                  55              3025                    0.954                           1
111111                  63              3969                    1.2517                          1
101011                  43              1849                    0.5831                          1
111110                  62              3844                    1.2122                          1

------------------------------------------ New Population 2 ------------------------------------------
Mate Pool       Mate            Crossover Points        New Population          x value        f(x)
------------------------------------------------------------------------------------------------------
110111          1               4                       110111                  55              3025
111111          0               4                       111111                  63              3969
101011          3               5                       101010                  42              1764
111110          2               5                       111111                  63              3969

------------------------------------------ GENERATION 3 ------------------------------------------
Initial Population      X Value         Fitness Value( f(x) )   Probability(Expected Count)     Actual Count
--------------------------------------------------------------------------------------------------
110111                  55              3025                    0.951                           1
111111                  63              3969                    1.2477                          1
101010                  42              1764                    0.5545                          1
111111                  63              3969                    1.2477                          1

------------------------------------------ New Population 3 ------------------------------------------
Mate Pool       Mate            Crossover Points        New Population          x value        f(x)
------------------------------------------------------------------------------------------------------
110111          3               3                       110111                  55              3025
111111          2               2                       111010                  58              3364
101010          1               2                       101111                  47              2209
111111          0               3                       111111                  63              3969
PS D:\SEM 5\AI\EXPERIMENTS>
```

Conclusion :

Thus we successfully studied and applied Genetic Algorithm