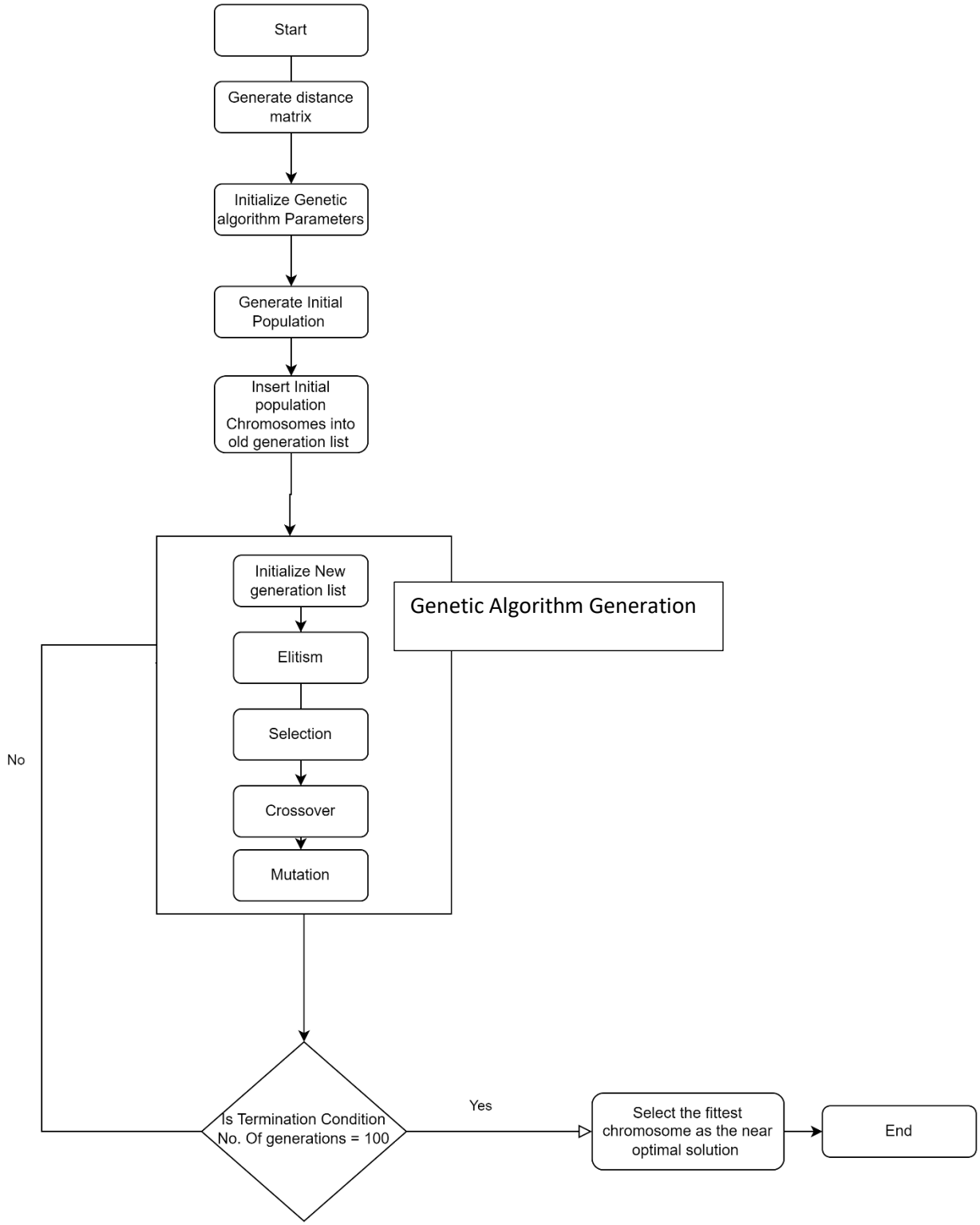
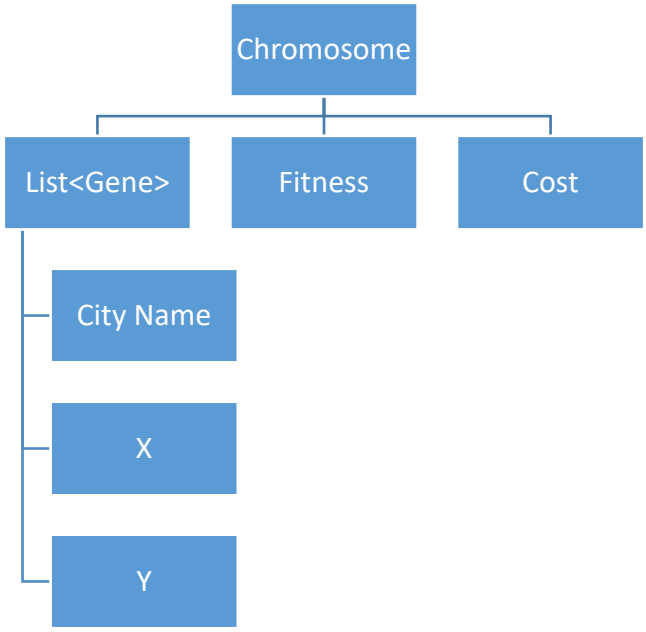
Genetic algorithm flowchart:



1- Genetic algorithm parameters:

a. Population size= 50
b. Generations Count=100
c. Elitism percentage = 2% of population (mean two chromosomes)
d. Crossover probability = 0.6
e. Mutation probability = 0.1

2- Chromosome structure



3-Initial Population

Function steps:

```
For(Initial population count)
{
    Generate Chromosome  From Permutation()
    Evaluate Chromosome Fitness()
    Add Chromosome To Population()
}
```

4-Fitness Evaluation

Fitness will be cost based

Equation:

Fitness = 1/ (Total cost of Tsp Trip)

Function steps:

a. Calculate Chromosome Cost ()
b. Set Cost to Chromosome
c. Calculate Chromosome Fitness(Cost)
d. Set Fitness To Chromosome

5-Elitism

Function steps:

a. Calculate Elitism Count ()
b. Select Top Fittest Chromosomes
c. Add Elite Chromosome To New Generation List

6-Selection **Hint Use K-Tournament Selection**

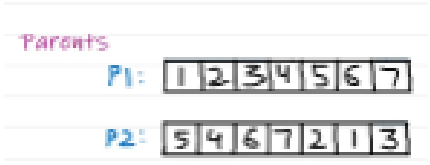
Function steps:

a. Select Random K Chromosomes ()
b. Chose the fittest()

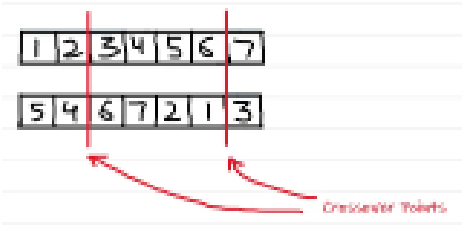
7-Crossover **Hint Use partially mapped Crossover**

Partially Mapped Crossover

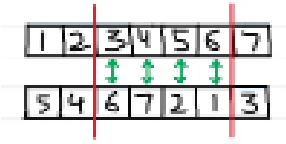
Step 1- Select two parents for mating.

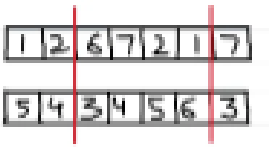


Step 2- Select a substring from the parents at random using crossover points.



Step 3- Perform Two-Point Crossover.

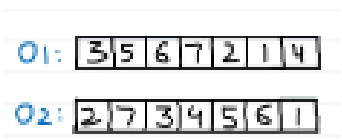




Step 4- Determine mapping relationship from substring.

Step 5- Use the mapping to legalize offsprings in crossbreed substrings.

After crossover, the new offsprings generated looks as follows:



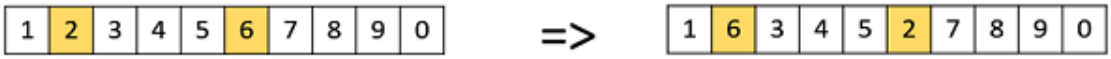
Function Steps:

a. Crossover Count =
   (population Size - No. of elite chromosomes) / 2
b. for(Crossover Count)
   { parent1= Tournament selection()
     parent2= Tournament selection()
   generate random double number that ranges from 0 ➔1
   if(generated random number < crossover probability){
     Apply Partial mapped crossover on parent1 and parent2
   Evaluate child1 and child2 Fitness
    insert child1 and child2 into new generation list
   }
   else {
   insert parent1 and parent2 into new generation list}
   }

8-Mutation **Hint use swap Mutation**

Swap Mutation

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings.



Function Steps:

a. for(population size)

{

parent1= select random parent from new generation list

generate random double number that ranges from 0 → 1

if(generated random number < mutation probability){

Apply swapping mutation on parent1

Evaluate mutated parent Fitness

replace parent1 by mutated parent into new generation list

}

}