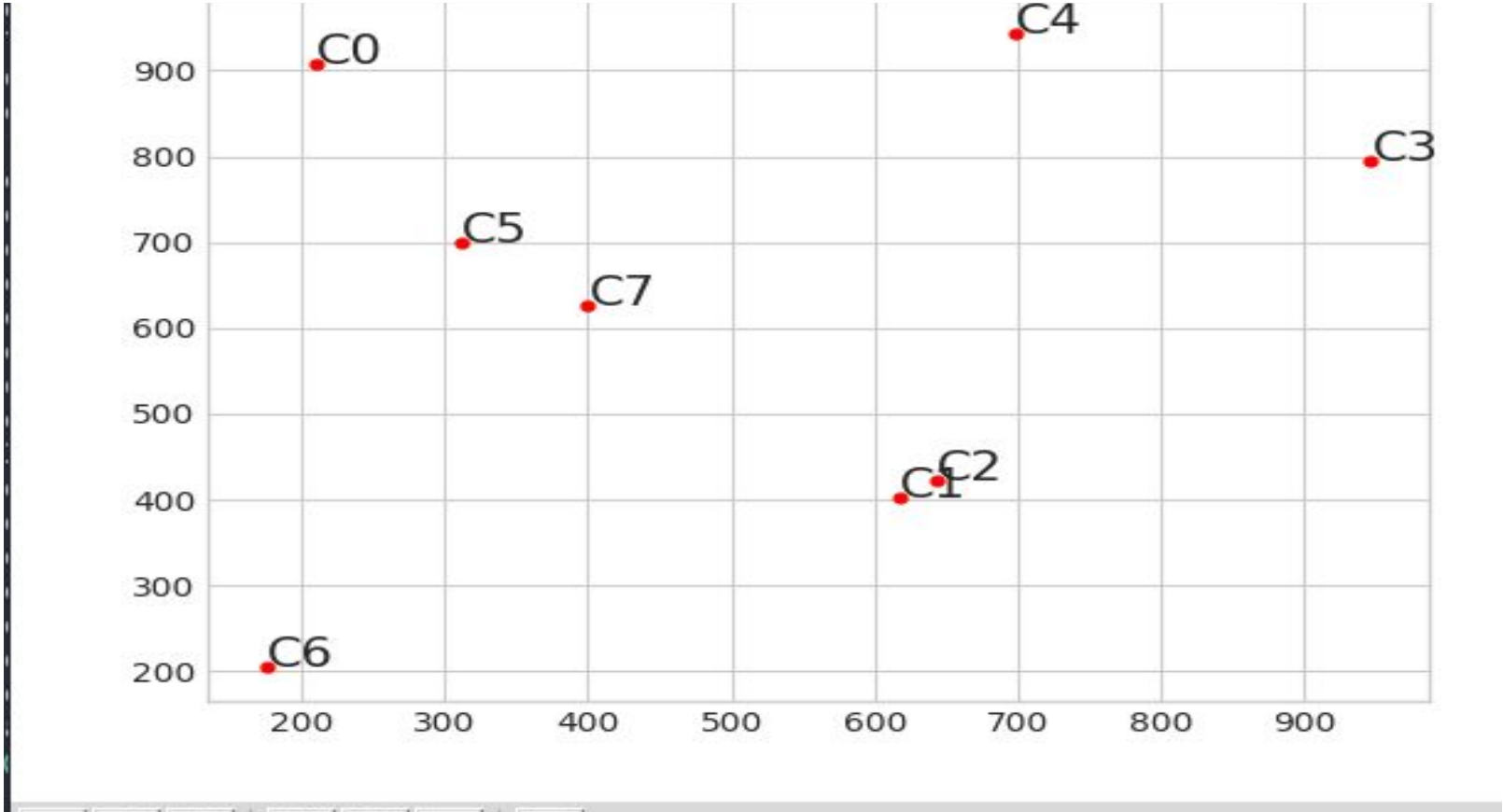
TSP solution via GA

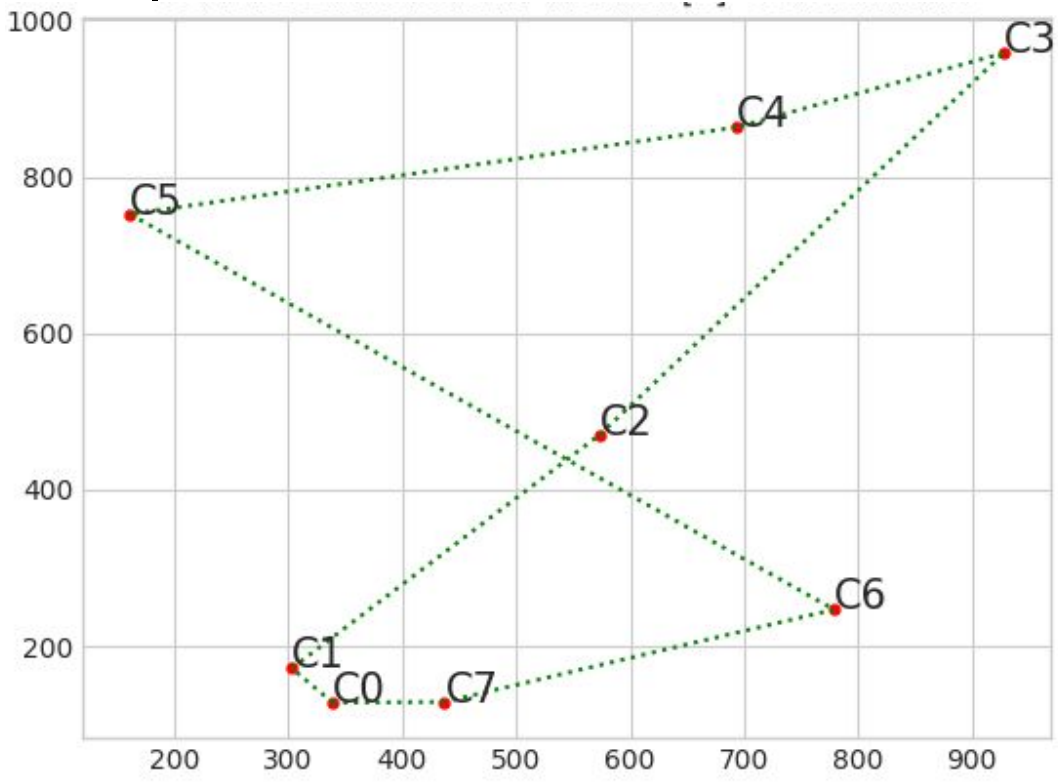
Jai Lulla B150394293
Hima Deo B150394235

The Problem


C0
C4
C3
C5
C7
C2
C1
C6
900
800
700
600
500
400
300
200
200
300
400
500
600
700
800
900

Individual | Chromosome| Candidate Solutions

- A path or a Candidate solution to the TSP problem
- Initial population is randomly assigned paths
- They may or may not have good fitness



Fitness Function

- 1/Summation(Euclidean Distance between consecutive cities)
- Euclidean Distance = (x1-x2)^2 - (y1-y2)^2
- Higher fitness better the solution | Individual

Encoding Used : Permutation Encoding

Order of traversal is order of array

Steps

- Initialize Population

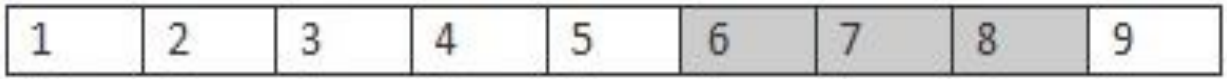
  Randomly generate candidate solutions| Chromosomes| Individuals

- Until Convergence Or generation limit
  1. Select Parents
  2. Crossover
  3. Mutate

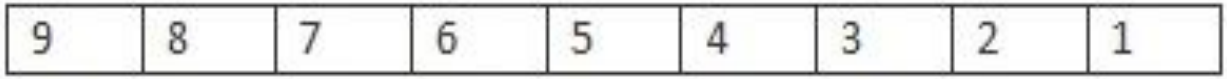
Different Methods For Selection Compared

1. Roulette Selection
2. Rank Selection
3. Tournament selection
4. Random Selection
5. Hybrid eg(Roulette-Rank)

Crossover



Parents



Child

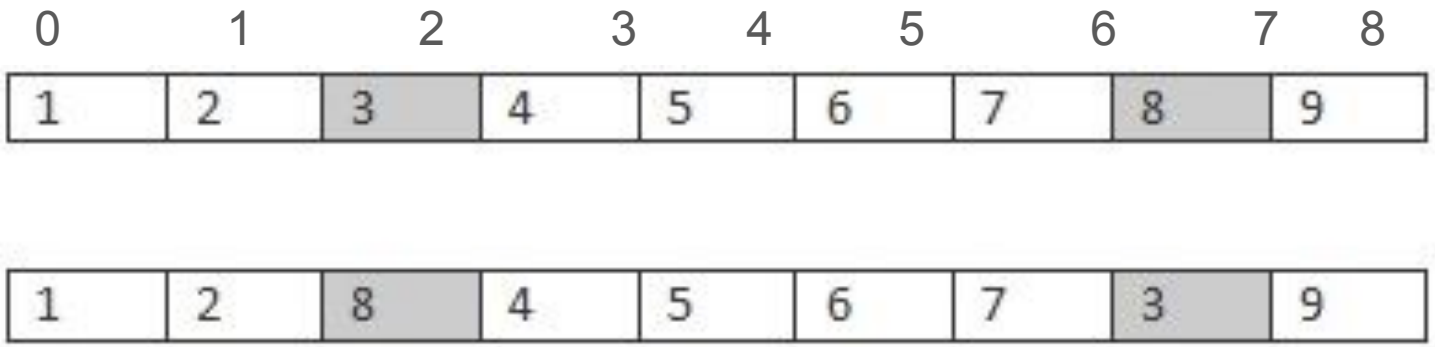
Step 1 :

Step 2:

Mutation : Swap mutation

Randomly Select two indices and Swap Cities

Always a proper solution to TSP as no cities are duplicated or repeated



Approach To collect data

- Run the code for “T” test cases (Problems | City locations)
- Try solving the same “T” cases one at a time using the different selection methods(eg Rank,Roulette..)
- Average out the fitness values over “T”
- Plot and Compare Results