Visualization of Traveling Salesman Problem with Genetic Algorithm and Ant Colony Optimization

Author: Shubhanshu Sharma (11943)

Affiliate: Cluster Innovation Centre, University of Delhi, 110007, India

Mentor: Dr. Nirmal Yadav

Introduction

Travelling Salesman Problem (TSP) a hypothetical salesman has to visit several cities, visiting each city only once, before ending the journey at the original starting city. The shortest path, or tour, of cities, is the solution to the problem among all possible tours.

Two methodologies and inbuilt functions are used to develop a visualization model for the problem:

Focus of Thesis

This study is conducted to determine which method amongst the brute-force, lexicographic order, genetic algorithm, or ant colony optimization is most feasible and optimized for solving the travelling salesperson problem.

❖ Ant Optimization Theory is used as a full-fledged model with all the pheromone trails from food sources and returning to hometown.

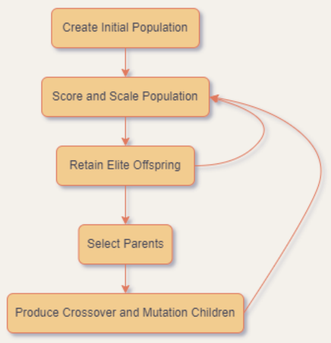
❖ The genetic Algorithm used in transportation networks will have an impact on the end output and journey time.

Methodology:

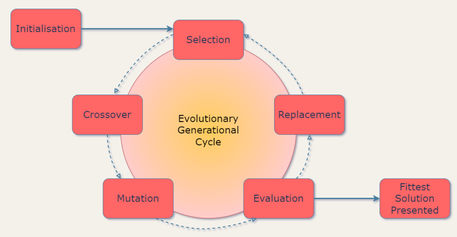
1. **Genetic Algorithm** optimization processes are based on the evolutionary principle of "survival of the fittest."

How the Genetic algorithm works:

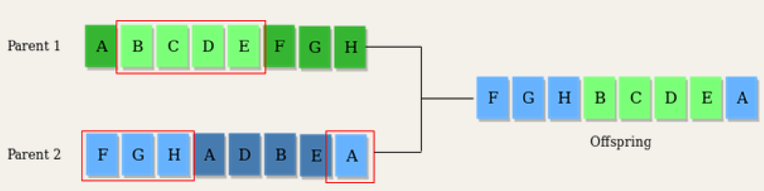
1 Flow-chart of genetic algorithm



2 Evolutionary Generational Cycle

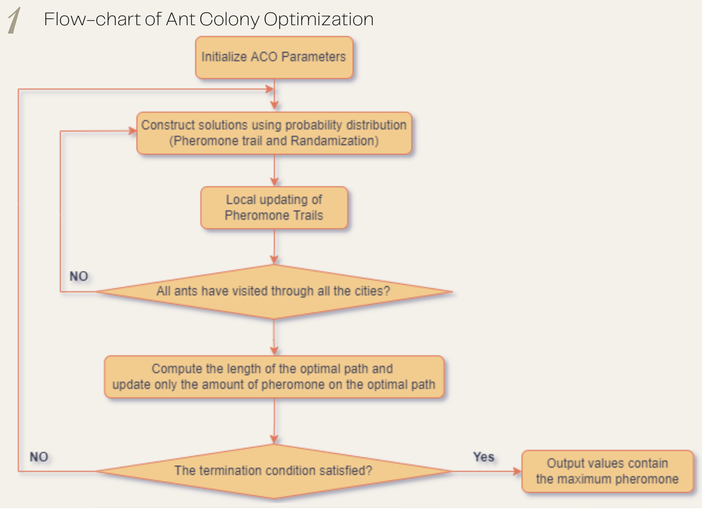


3 How the offspring is created using the genes of parents.



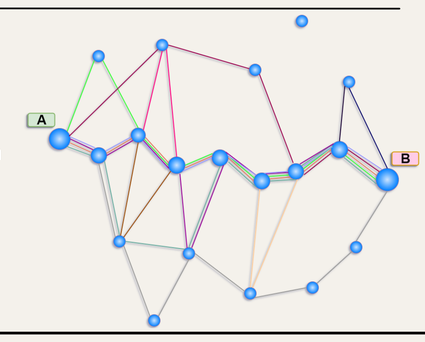
2. **Ant colony** optimization is based on ant movement theory which is inspired by the 'Foraging' behaviour of ants.

1 Flow-chart of Ant Colony Optimization



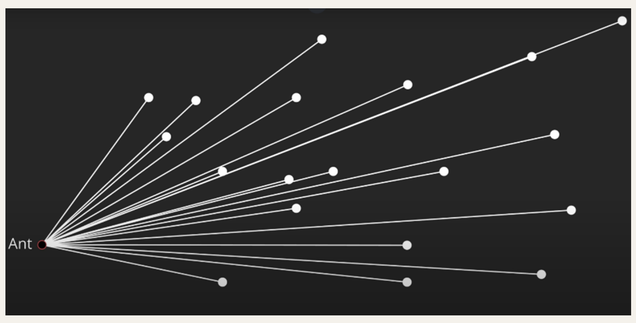
2

Bunch of ants are traversing through several towns to reach from town A to town B



Implementation Detail: How does **Ant Colony Optimization** work?

1. **Placing a virtual** ant in our program which will pick the nearest town by the formula of desirability that is *(1/Distance)* ^{Power}



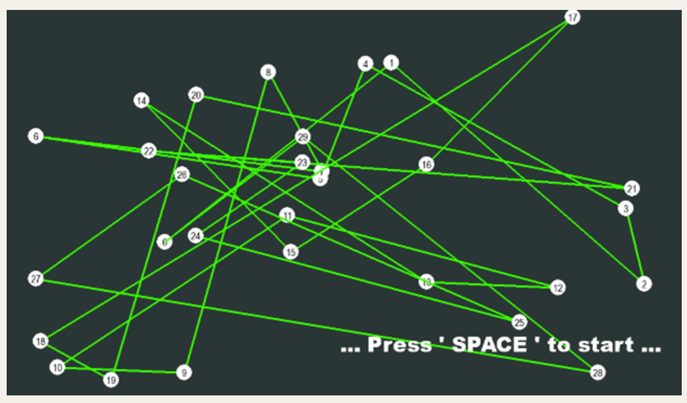
Ant picking the nearest town having highest desirability.

2. **Creating an ant**: While creating an ant we will introduce 3 important parameters in it, these are:

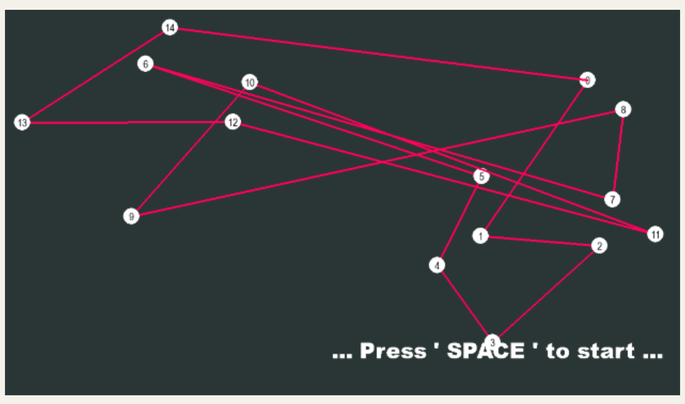
- Alpha – This parameter is going to control the importance of the pheromone trail
- Beta – This parameter is going to control the heuristic information during the selection of the pheromone trail.
- State – Is a point on a graph or a town.

3. **Creating towns**:

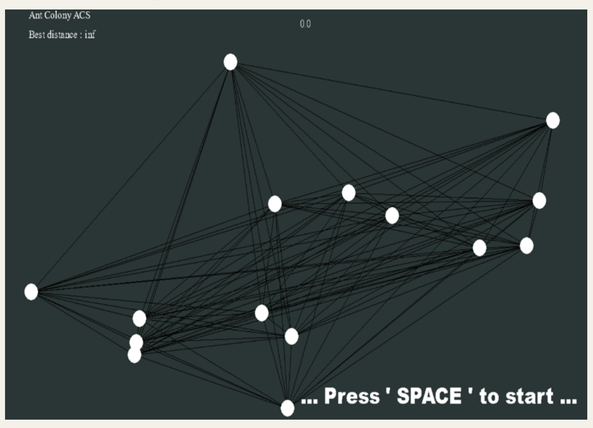
- A 30 town map.



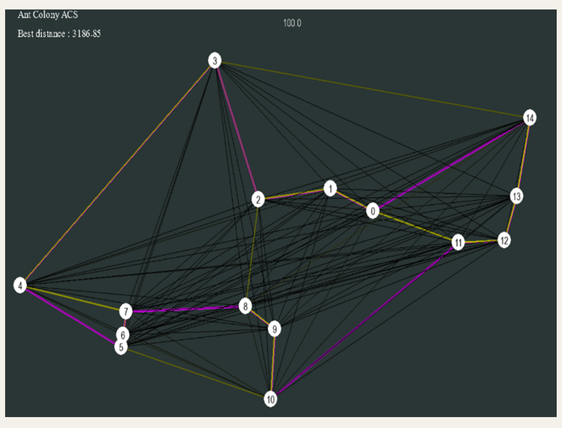
- A 15 town map.



4. **Implementing ants in an Ant Colony**: All we need is to add pheromones and a method to deploy our ants on our map. Finally, we will draw the final path traversed by our little ants.



Generating a map for ACS simulation



Simulation finished, Path generated is shown by bright purple and orange color

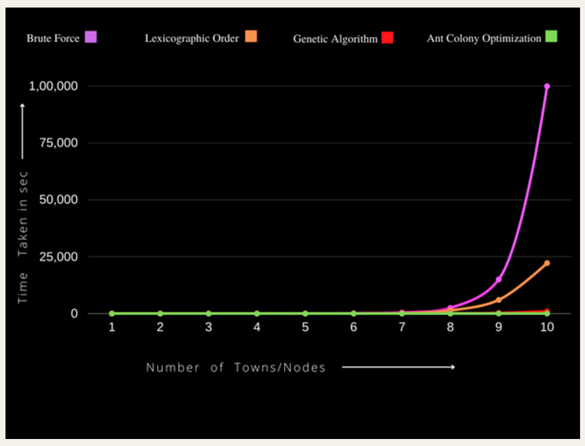
Results:

After running and defining all the algorithms the final results show us the best way to solve the travelling salesperson problem is Ant Colony Optimization based on ant movement theory.

The following graph will give us the results of the time taken by the algorithms:

	Brute force	Lexicographic	Genetic	ACO
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0.5	0.05	0	0
6	150	2	0.5	0.1
7	500	171	1	0.1
8	2500	1350	100	5
9	15000	7000	300	11
10	100000	35000	1000	11.1

Values generated by the program (number of towns vs time taken in seconds)



Comparison all algorithms used for solving traveling salesperson problem

Future Directions:

- A model that can be created is used to manage the inventory in an industry or a warehouse. For example, Walmart the biggest multinational retail corporation can use this model to manage the inventory in their warehouses.
- To optimize the travel time for the transportation of products from the warehouse to the consumers. For example, Amazon can use this optimization technique to deliver the product faster to their customers.

Refrences:

https://www.researchgate.net/publication/341371861_Literature_Review_on_Travelling_Salesman_Problem
https://natureofcode.com/book/chapter-9-the-evolution-of-code/
http://graphics.cs.uh.edu/wp-content/papers/2018/2018-CAVW-AntSimulation.pdf
https://ieeexplore.ieee.org/document/5429417