Soft Computing (INT246) Project Report

Travelling Salesman Problem using Ant Colony Optimization and Genetic Algorithm

Section: **KM107**

Submitted to: **Dr Aditya Khamparia**

Submitted by:

Sr. No.	Registration No	Name of Student	Roll No
1	11803483	Bhupesh Parakh	A14
2	11810004	Resham Chaney	A24
3	11803909	Heena Rawat	A33
4	11912827	Twinkle Gupta	B60

Acknowledgement

I would like to express my gratitude towards my University as well as teacher for providing me the golden opportunity to do this wonderful project , which also helped me in doing a lot of homework and learning. As a result, I came to know about so many new things. So, I am really thank full to them.

Moreover I would like to thank my friends who helped me a lot whenever I got stuck in some problem related to my course. I am really thankfull to have such a good support of them as they always have my back whenever I need.

Also,I would like to mention the support system and consideration of my parents who have always been there in my life to make me choose right thing and oppose the wrong. Without them I could never had learned and became a person who I am now.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

Twinkle Gupta
11912827

Heena Rawat
11803909

Bhupesh Parakh
11803483

Resham Chaney
11810004

Introduction

The Travelling Salesman Problem (TSP) is a complex problem in combinatorial optimization that cannot be solved conventionally particularly when the number of cities increases. The aim of this project is to compare the effect of using two distributed algorithms (popular meta-heuristics techniques used for optimization tasks) which are Ant Colony Optimization (ACO) as a Swarm intelligence algorithm and Genetic Algorithm (GA).

For Ant Colony Optimization, we studied the effect of some parameters on the produced results, these parameters as: number of used ants, maximum tours, and number of cities. On the other hand, we studied the chromosome population, crossover probability, and mutation probability parameters as well as the effect of number of cities, generations and population effect on the Genetic Algorithm results. The comparison between Genetic Algorithm and Ant Colony Optimization is accomplished to state the better one for travelling salesman problem. The results of comparison show that ant colony is better than genetic algorithm and it generally requires only a few lines of code.

Travelling salesman problem

The traveling salesman problem (TSP) is one of the well-known problem in discrete or combinational optimization. A traveling salesman problem is the problem in which a sales man has to visit all the cities exactly ones into a given number of city set and come back to the home city in such a way the total cost of visiting the cities should be minimum. TSP is an NP-Hard problem which means there is no particular solution for solving the TSP and it is very easy to describe but so difficult to solve.
A general Travelling Salesman Problem can be represented by a complete weighted directed graph G = (V, E), where V = {1, 2, 3....n} is a set of vertices representing cities and E being the set of arcs fully connecting the nodes [8]. In addition, E is associated with the set D representing costs d, where dij is the metric of the distance between the cities i and j. dij can be defined within the Euclidean space and is given as follows:

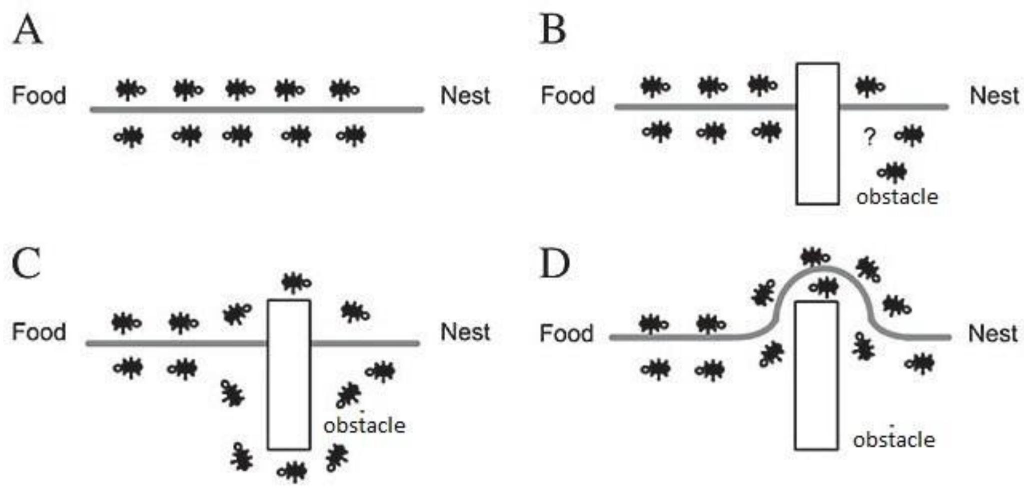
$$d = \surd(x - x\ ) + (y - y\ )$$

The objective of the TSP is to find shortest closed tour in V such that each city is visited once and only once.

Solving TSP using ACO and GA

ANT COLONY OPTIMIZATION

The ant colony optimization first introduced in 1991 by A. Colorni M. Dorigo, and V. Mahiezzo. The algorithm is simulated with the behavior of the real ants. The real ants having the capability to find the shortest path from the food source to the nest (destination) without any visual cues. Ants can communicate with each other through chemical substance called pheromones in their immediate environment also, they are capable of adapting to change in the environment, for example finding a new shortest path once the old one is no longer feasible due to a new obstacle. The ants deposit some amount of pheromones on the ground while walking from nest to the food source or food source to the nest or vice versa this pheromone is the trivial factor for finding the best solution. An ant sense the pheromone deposited by the other (earlier) ants and tend to imitate the trail with stronger pheromone. So a shorter path having a higher amount of pheromone is probability and ants will tend to choose the shorter path.



Approach

The proposed approach is to generate the population randomly. Place the ant at any particular city randomly. The probability to choose the next city is generated depending upon the amount of pheromone and the heuristic factor. Next city is chosen based on the probability (roulette wheel selection). The tour and the distance travelled is saved. The pheromone is updated for each ant. It is compared with the global tour and the best between them is selected as global best ant. At the end of each iteration the pheromone is evaporated. The same process is continued for certain number of iterations.

Algorithm used:

```
Begin
    Initialize the cities (nodes) and the pheromone trail
        Generate the population of the given colony size
        Generate the cost(distance) between the nodes.
        For each individual ant
            Find the tour and the total distance and add pheromone in the path.
        Determine the best global ant and add pheromone in the path of the best ant.
        Evaporate the pheromone.
    Check if termination = true
End;
```

Probability calculation formula:

An ant will move from node i to node j with probability

$$p_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})|}$$

where
$\tau_{i,j}$ is the amount of pheromone on edge i, j
α is a parameter to control the influence of $\tau_{i,j}$
$\eta_{i,j}$ is the desirability of edge i, j (typically $1/d_{i,j}$)
β is a parameter to control the influence of $\eta_{i,j}$

Pheromone update formula:

Amount of pheromone is updated according to the equation

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}$$

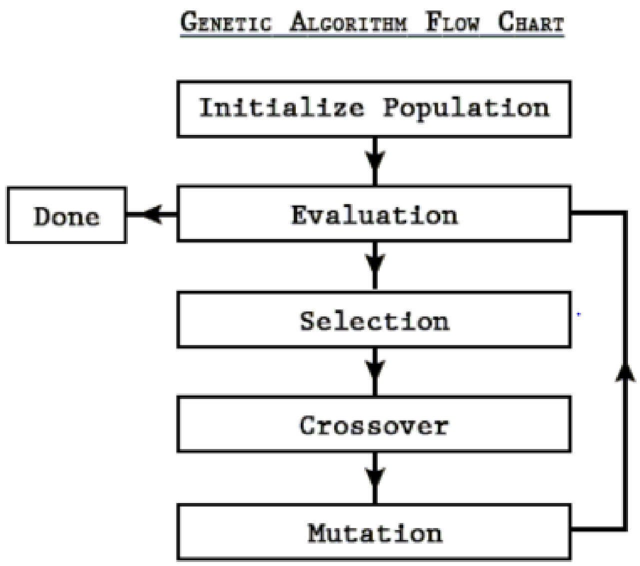
where
$\tau_{i,j}$ is the amount of pheromone on a given edge i, j
ρ is the rate of pheromone evaporation
$\Delta\tau_{i,j}$ is the amount of pheromone deposited, typically given by

$$\Delta\tau_{i,j}^{k} = \begin{cases} 1/L_k & \text{if ant } k \text{ travels on edge } i, j \\ 0 & \text{otherwise} \end{cases}$$

where L_k is the cost of the k^{th} ant's tour (typically length).

GENETIC ALGORITHM

GA's have been around since 1957, starting with simulations for biological evolution. GA's are used for optimization problems with large search spaces. The TSP as an optimization problem therefore fits the usage and an application of GA's to the TSP was conceivable. In 1975 Holland laid the foundation for the success and the resulting interest in GA's. With his fundamental theorem of genetic algorithm, he proclaimed the efficiency of GA's for optimization problems. A generic GA starts with the generation of a population of several different tours. Two tours (parents) are combined with crossover, the conceived child may experience mutation and is then given a fitness-score, in case of the TSP the tour length. On basis of the fitness-score a part of the population is discarded and the rest is used to create new offspring. GA's are tightly interwoven with local search, together they perform better than local search alone, this is because local search loses itself in local optima. GA's have the inherently good capability to merge tours and develop a better tour from good sub tours.

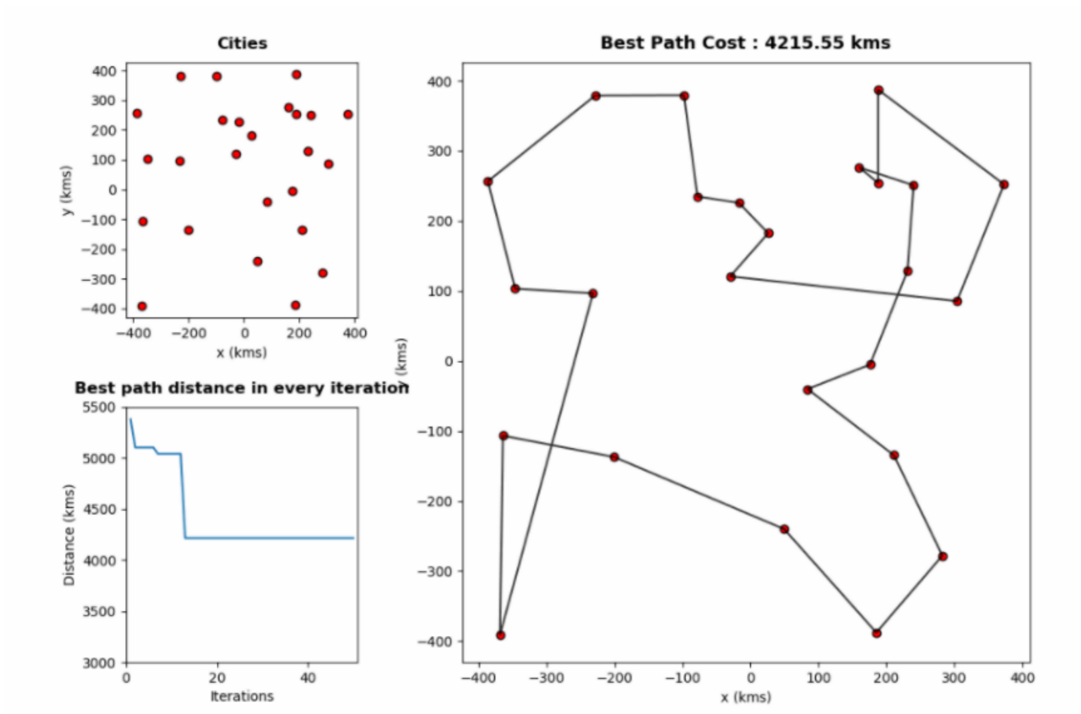


Algorithm:

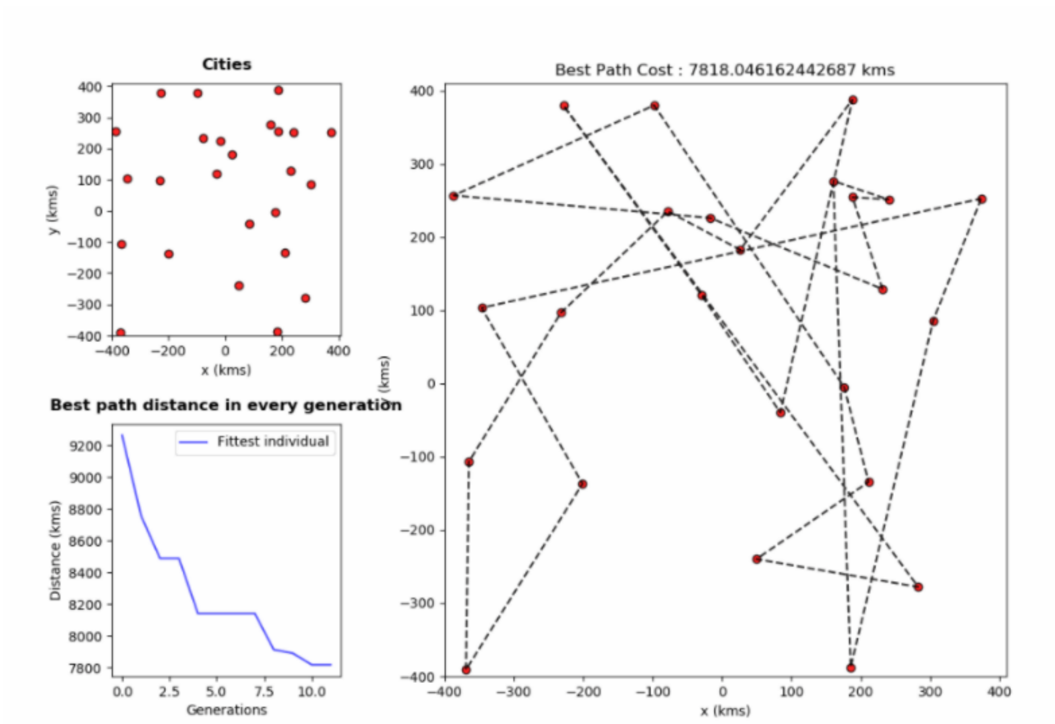
```
Begin
  initialize population
  find fitness (shortest distance) of population
  while (last generation is reached) do
    parent selection based on individuals' aptitude
    multipoint crossover with probability pc
    mutation (swap mutation) with probability pm
    decode and fitness calculation
    survivor selection
    find best
  return best
```

Results

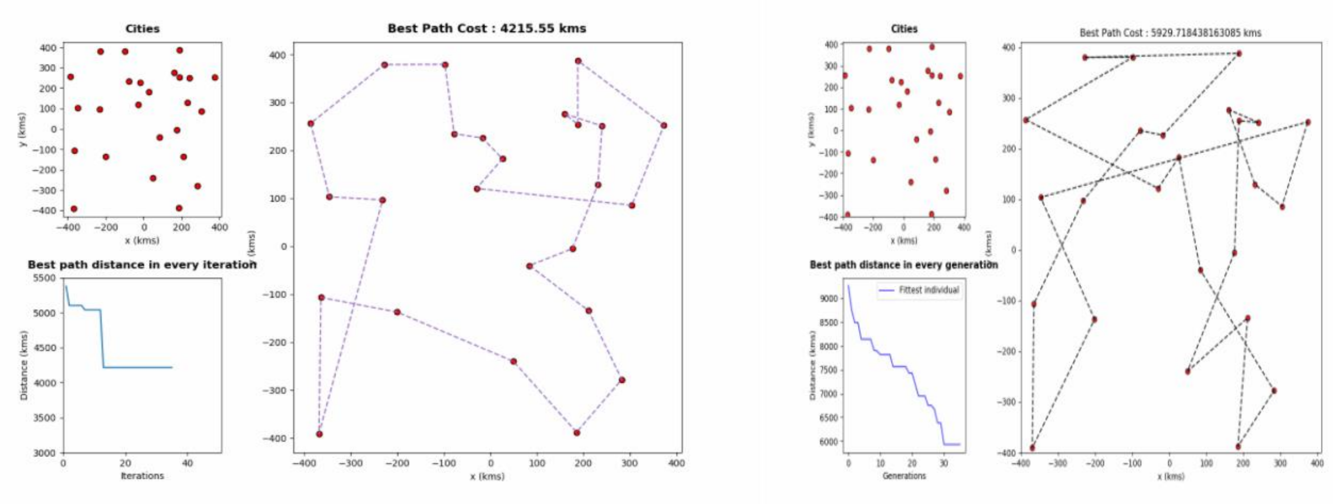
Ant Colony Optimization



Genetic Algorithm



Comparison of the two approaches



Conclusion

This represents an efficient approach for solving travelling salesman problem based on the divide and conquers strategy

(in which cities are divided into the number of sub city) in which the ant colony algorithm using the candidate list is applied. From our experimental result the proposed algorithm is more effective than the conventional ant colony algorithm to find the better solution. The development od genetic algorithm has been overloaded from beginning to present. A shift from inherent GA operation to more and more problem specific knowledge exploitation is visible. The mayor milestones are highlighted and the timespan has been divided into 3 epochs. These periods show distinct features in the areas participation, performance gains and refinement of implementations. In the first phase inception a barrage of different ideas is introduced ,of which few stand the test of time, implementations are simple and greater performance leaps are common. In the phase two improvement the proven good solutions lay the groundwork for more sophisticated algorithm that makes small improvements or enhance only edge-case of the problem. Ground breaking new discoveries on which can be build upon are rare. In the current third phase maturity is in immensely difficult to find the better algorithm, opportunities for further optimization may lay in specialized hardware, but the interest and therefore scientific effort starts to decline. The reason for the decline in appeal of GA’s seems to be that the field has been extensively studied and the investment of time and money doesn’t seems to be worth it to find a just marginally better algorithm.

References

1.wikipedia

2.www.github.com

3.www.slideshare.com