



# [Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator]

---

Report by Munir Hassan (k163761), Basit Ali Khatri (k163762) and Malik Obaid-Ur-Rehman (k163767)

SECTION: D

# Travel Salesman Problem

## Abstract

Genetic algorithms are evolutionary techniques used for optimization purposes according to survival of the fittest idea. These methods do not ensure an optimal solution; however, they give good approximation usually in a time. The genetic algorithms are useful for NP-hard problems, especially the traveling salesman problem. The genetic algorithm depends on selection criteria, crossover and mutation operators. To tackle the traveling salesman problem using genetic algorithms, there are various representations such as: binary, path, adjacency, ordinal and matrix representations. In this article, we propose a new crossover operator for traveling salesman problem to minimize the total distance. This approach has linked with path representation, which is the most natural way to represents a legal tour. Computational results are also reported with some traditional path representation methods like partially-mapped and order crossovers along with new cycle crossover operator for some benchmark TSPLIB instances and found improvements.

## Introduction

The Traveling Salesman problem (TSP) is one of the benchmarks and old problems in Computer Science and Operations Research. It can be stated as:

A network with 'n' nodes (or cities), with 'node 1' as 'headquarters' and a travel cost (or distance, or travel time etc.,) matrix  $C = [c_{ij}]$  of order n associated with ordered node pairs (i, j) is given. The problem is to find a least cost Hamiltonian cycle.

On the basis of the structure of the cost matrix, the TSPs are classified into two groups – symmetric and asymmetric. The TSP is symmetric if  $c_{ij} = c_{ji}$ ,  $\forall i, j$  and asymmetric otherwise. For an n – city asymmetric TSP, there are  $(n - 1)!$  possible solutions, one or more of which gives the minimum cost.

For an n-city symmetric TSP, there are  $\frac{(n - 1)!}{2}$  possible solutions along with their reverse cyclic permutations having the same total cost. In either case the number of solutions becomes extremely large for even moderately large n so that an exhaustive search is impracticable.

## Genetic Algorithm

Genetic algorithms (GAs) are derivative-free stochastic approach based on biological evolutionary processes proposed by Holland. In nature, the most suitable individuals are likely to survive and mate; therefore, the next generation should be healthier and fitter than previous one. A lot of work and applications have been done about GAs in a frequently cited book by Goldberg. GAs work with population of chromosomes that are represented by some underlying parameters set codes.

## Design

1. First, our program populates array of cities through which the salesman will travel.
2. Our program makes orders (chromosomes)
3. **Chromosomes:** it is a specific order in which the salesman will travel.
4. It will calculate total distance of each chromosomes.
5. It will populate many random chromosomes.
6. Then it will calculate the fitness of each item in the population array.
7. It will normalize the fitness.
8. Now it will continue generating new populations:
  - o Find two parents.
  - o Apply Crossover function on them.
  - o It will generate two more population.
  - o Mutate these population.
  - o Repeat the process till we get the 100% fitness.
9. Draw graphs on web page:
  - o Present Best Path
  - o Best Path.
10. We will get our final solution in the best path graph.

## Experimental Setup

### Required Inputs:

1. Total cities to visit by the salesman
2. Population size
3. Mutation rate

**Note:** You can change input by navigating in drawinOnBrowser.js file

### Languages Used:

1. HTML
2. JavaScript
3. P5 Library for drawing and array manipulation

**Note:** Internet connection is mandatory because we are using p5 library.

## References

P5 library:

- <https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.6.1/p5.js>
- <https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.6.1/addons/p5.dom.js>

## Code

1. Four files (1 HTML, 3 JavaScript)
2. drawingOnBrowser.js contains every function of drawing the graph.
3. GeneticAlgorithmBasics.js contain every function of GA like mutate, crossover etc.
4. Utility.js contain Utility functions.

**Note:** All the code is fully commented.