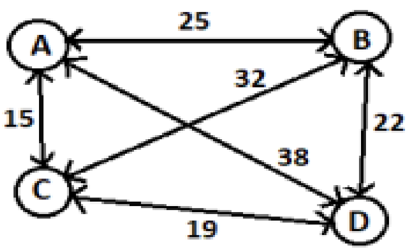
Travelling Salesman Problem With Genetic Algorithm

Midnapore College (Autonomous), BCA-5TH Sem.-(Project)
Group : Suman Paik | Arun Sarma | Rohan Das | Sumit Mondal

Abstract - The Traveling Salesman Problem (TSP) isa classical combinatorial optimization problem, which is simple to state but very difficult to solve. The problem is to find the shortest tour through a set of N vertices so that each vertex is visited exactly once.In this paper we provide overview of GeneticAlgorithm Which is used for solving the travelling salesman problem.We present crossover and mutation operators, developed to tackle the Travelling Salesman Problem with Genetic Algorithms with different representations.we show the experimental results obtained with different standard examples using combination of crossover and mutation operators in relation with path representation.

Introduction:- The Travelling Salesman Problem is a classical combinatorial optimization tour problem in which given a list of cities and the distance of travel between each pair of cities and each city make exactly one visit. TSP is to find the shortest way of visiting all cities and return to the start point. Although the statement is easy to say, it is more difficult to resolve. Travelling Salesman Problem is an optimization problem and there is a vast search space and is called NP-hard, which means it cannot be solved in polynomial time. It is one of the most fundamental problems in computer science today. We knowTSP first described the Irish and British mathematicians WR Hamilton and Thomas Kirkman in the 1800s by creating a game that was solvable by finding a Hamilton cycle, a non-overlapping path between all nodes. TSP is being applied in many cases nowadays. These are microchip making, vehicle routing, drilling on printed circuit boards, Overhauling gas turbine engines etc. Simply put, we have a set of numbers of n cities, and then we can get (n - 1)! alternative routes to cover all n cities. With the help of TSP, collect the route that has the least distance.

Let us consider that we have a set of five cities A, B, C and D. Distances between cities are given us. Here the route can be generated (4-1)! means 6.



In this figure , all tour and cost->

1) A - B - D - C - A -> 25 + 22 + 19 + 15 =81
2) A - B - C - D - A ->25 + 32 + 19 + 38 =114
3) A - C - D - B - A -> 15 + 19 + 22 + 25 = 81
4) A - C - B - D - A -> 15 + 32 + 22 + 38 = 107
5) A - D - B - C - A -> 38 + 22 + 32 + 15 = 107
6) A - D - C - B - A -> 38 + 19 + 32 + 25 = 114

Here we can clearly see that no.1 and no.3 take the minimum cost to travel between each city and go back to its beginning. Many heuristic techniques have been used to find efficient solutions to problems such as greedy method, ant algorithm, simulated annealing, taboo search and genetic algorithm. But as the number of cities increased, it became difficult to find solutions to the calculations. Despite the computational difficulties, we can use methods like genetic algorithms that can come close to the best solution for thousands of cities.

TSP Variations:

Asymmetric and symmetric TSP:-

There are two main forms of TSP - symmetric TSP (STSP) and asymmetric TSP (ATSP).In the case of STSP, the distance between the two cities is the same on each opposite side, so also the cost. In ATSP, the distance between two cities is different when travelling in the opposite direction; Costs may vary; at times only one-way traffic is allowed on the way from one city to another.TSP is relatively easier to solve than ATSP, since symmetry halves the number of possible solutions.

The Generalized Travelling Salesman Problem (GTSP):-

The GTSP is a form of standard travelling salesman problem. Like TSP, the considered graph consists of n nodes and the cost is known between any two nodes. An optimal GTSP solution is a minimum cost cycle that visits exactly one node from each cluster.

Different solutions for TSP:

Naive Solution:-

i. Consider city 1 as the starting and ending point.
ii. Generate all (n-1)! Permutations of cities.
iii. Calculate cost of every permutation and keep track of minimum cost permutation.
iv. Return the permutation with minimum cost. Time Complexity: (n!)

Various Method For Solving TSP:

Method of total enumeration:-

The method is the total number of all possible routes (n-1)!. If the number of places visited is high, it is not employable. The amount of potential solutions increases rapidly with each additional element(nodes)and even nowadays our computer is not strong enough to be able to provide the best solution in a reasonable time. So this method is not efficient for tsp problems.

Brute force Approach:-

Many problem solving methods were used to solve the TSP problem. One of the well-known and straightforward approaches is the brute force approach. This method gives approximate time to solve tsp with 16 nodes in more than 2 years. The time taken to solve cities with more time is unexpected, there are 2n to find the right solution to the n size problem. n Sub problems need to be solved in linear time. It takes O (2^n.n^ 2). So it cannot be applied for TSP.

Nearest Neighbourhood:-

The Nearest neighbourhood method looks at the distances of nearby cities that have not been visited and returns to the starting city when all other cities are visited. It takes less time to find a solution using this method than the brute force method, but this method does not guarantee the best solution as a brute force approach. This way the salesman can start in any city. This algorithm is very easy to implement, but in the worst case it creates a solution that is much longer than the optimal solution.

Greedy approach:-

This is the first method of solving TSP. This method uses the protruding edges from the node and selects the edge with the lowest cost. If these n cheap edges form a Hamiltonian cycle then we can get an optimal solution. The greedy algorithm is not easy to complete the lower bound by providing the upper bound. It will create the optimal local solution, but does not guarantee the optimal global solution.

Branches and bounds:-

The Branch and bound technique is most commonly used in travelling salesman problems by creating a state space tree to find the best solution out of all possible solutions with the value of the objective function. The branch and bound techniques give all possible solutions by solving problems, trying practical solutions and starting values at the upper bound to find the optimal solution but very laborious and does not always guarantee the optimal solution on the first try.

Computer Simulation:-

The development of simulation models, their program support and increasing computing power have led to efforts to use simulation techniques to solve large TSPs. Their qualifications are largely in random PC sampling which is then evaluated according to a selected objective function. Although the solution does not guarantee the global optimum, a sufficient number of simulations will be issued to achieve the best possible solution, the value of which will be close to the optimum.

Ant Colony Optimization (ACO):-

Researchers are often trying to mimic nature to solve complex problems, and one such example is the successful use of ACO. The idea is to mimic the movement of ants. This concept has been quite successful when applied to TSP, quickly giving the best solution to small problems (Dorigo and Gambardella).However, as small as an ant's brain may be, it is still much more complicated to mimic it completely. But we need a small part of their behavior to solve the problem. Pheromone tails leave when ants explore new areas. This is the way to direct other ants to potential food sources. The key to ants' success is strength in numbers, and the same goes for ant colony optimization. We start with a group of ants, usually 20 or more. They are placed in random cities and then told to move to another city. They are not allowed to enter any of the cities visited by themselves until they finish our tour. The ant that chooses the shortest tour will leave a tail of pheromones inversely proportional to the length of the tour. This pheromone trail will be considered when an ant chooses a city to go, creating a tendency to walk the path with the strongest pheromone trail. This process is repeated until a tour is found short enough. Ant Colony Optimization has been used to create near-optimal solutions to the Travelling Salesman problem.ACO is able to search the global optimum in a finite time. But this algorithm does not give fast results.

Genetic Algorithm :-

A genetic algorithm is a heuristic search algorithm used to solve search and optimization problems. This algorithm can solve problems very fast which if you try to solve by brute force that would take too much time for it. This algorithm can be used in many real-life applications such as data centers, image processing, codebreaking, electronic circuit design, and artificial creativity. This algorithm is a subset of evolutionary algorithms, which are used in computation. Genetic algorithms are inspired by the concept of genetics and natural selection to provide best solutions to problems. GA is based on the behaviour of chromosomes and their genetic structure. Every chromosome plays the role of providing a possible solution. The fitness function helps in providing the characteristics of all individuals within the population. The greater the function, the better the solution. That's why these algorithms have better intelligence than random search algorithms because these algorithms use historical data to take the search to the best performing region within the solution space.

Basic Terminologies -

Population: This is a subset of all the probable solutions that can solve the given problem.
Chromosomes: A chromosome is one of the solutions in the population.
Gene: This is an element in a chromosome.

Allele: This is the value given to a gene in a specific chromosome.

Fitness function: This is a function that uses a specific input to produce an improved output. The solution is used as the input while the output is in the form of solution suitability.

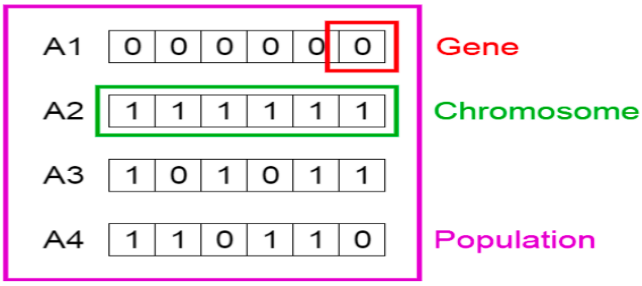
Genetic operators: In genetic algorithms, the best individuals mate to reproduce an offspring that is better than the parents. Genetic operators are used for changing the genetic composition of this next generation.

Advantages of Genetic Algorithm:-

1) It has excellent parallel capabilities.
2) It can optimize various problems such as travelling salesman problems ,discrete functions, multiobjective problems, and continuous functions.
3) It provides answers that improve over time, meaning the generated solution set improves by each generation.
4) A genetic algorithm does not need derivative information.

Genetic Algorithm Working Model :-

Genetic algorithms use the evolutionary generational cycle to produce high-quality solutions. They use many different operations that increase or replace the population to provide an improved fit solution.



Genetic algorithms follow the following phases to solve complex optimization problems:

Initialization:-

The genetic algorithm starts by generating an initial population. This initial population consists of all the probable solutions to the given problem. The most popular technique for initialization is the use of random binary strings.

Fitness assignment:-

The fitness function helps in establishing the fitness of all individuals in the population. It assigns a fitness score to every individual, which further determines the probability of being chosen for reproduction. The higher the fitness score, the higher the chances of being chosen for reproduction.

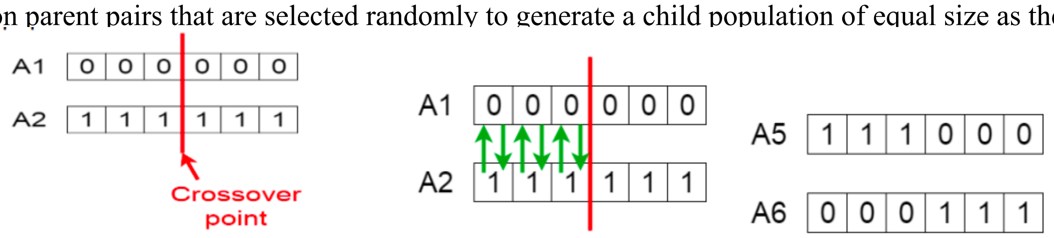
Selection:-

In this phase, individuals are selected for the reproduction of offspring. The selected individuals are then arranged in pairs of two to enhance reproduction. These individuals pass on their genes to the next generation. The main objective of this phase is to establish the region with high chances of generating the best solution to the problem (better than the previous generation). The genetic algorithm uses the fitness proportionate selection technique to ensure that useful solutions are used for recombination.

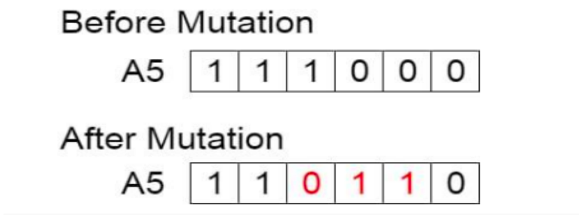
Reproduction:

This phase involves the creation of a child population. The algorithm employs variation operators that are applied to the parent population. The two main operators in this phase include crossover and mutation.

1) Crossover:- This operator swaps the genetic information of two parents to reproduce an offspring. It is performed on parent pairs that are selected randomly to generate a child population of equal size as the parent population.



2) Mutation:- This operator adds new genetic information to the new child population. This is achieved by flipping some bits in the chromosome. Mutation solves the problem of local minimum and enhances diversification. The following image shows how mutation is done



3)Replacement:- Generational replacement takes place in this phase, which is a replacement of the old population with the new child population. The new population consists of higher fitness scores than the old population, which is an indication that an improved solution has been generated.

4)Termination:- After replacement has been done, a stopping criterion is used to provide the basis for termination. The algorithm will terminate after the threshold fitness solution has been attained. It will identify this solution as the best solution in the population.

Genetic Algorithm use cases:

1) Transport:-

Genetic algorithms are used in the traveling salesman problem to develop transport plans that reduce the cost of travel and the time taken. They are also used to develop an efficient way of delivering products.

2) DNA Analysis:-

They are used in DNA analysis to establish the DNA structure using spectrometric information.

3) Multimodal Optimization:-

They are used to provide multiple optimum solutions in multimodal optimization problems.

4) Aircraft Design:-

They are used to develop parametric aircraft designs. The parameters of the aircraft are modified and upgraded to provide better designs.

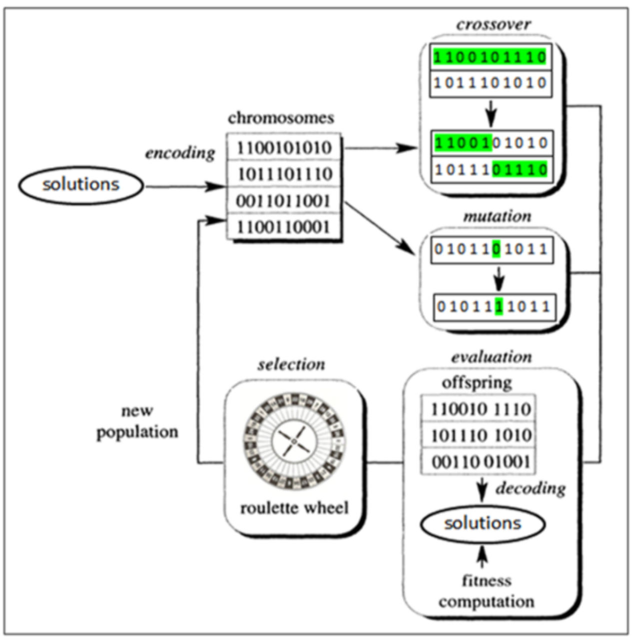
5) Economics:-

They are used in economics to describe various models such as the game theory, cobweb model, asset pricing, and schedule optimization.

The pseudocode of Genetic Algorithm :

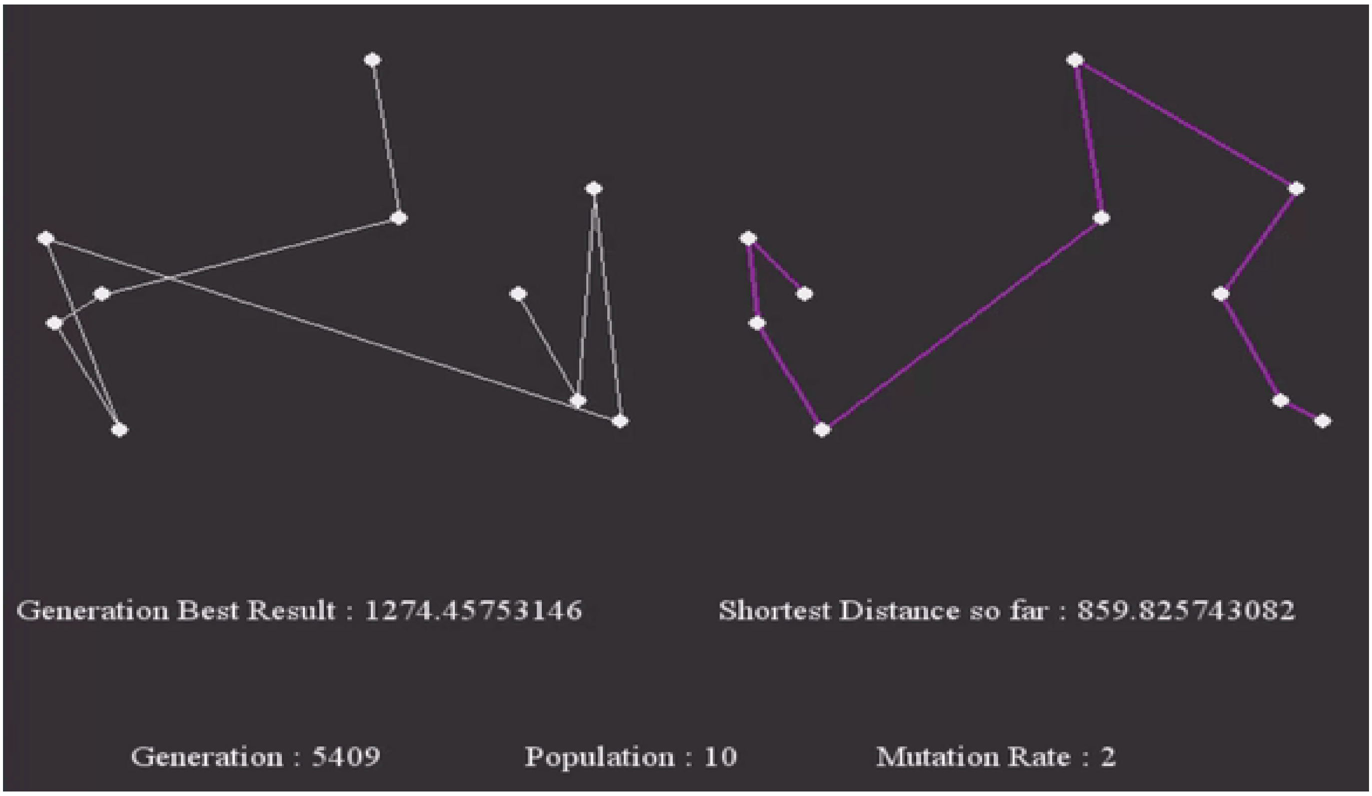
BEGIN GA
Make the initial population randomly.
WHILE NOT stop **DO**
BEGIN
Select parents from the population.
Produce children from the selected parents.
Mutate the individuals.
Extend the population adding the children to i
Reduce the extended population.
END

Output the best individual found.
END GA

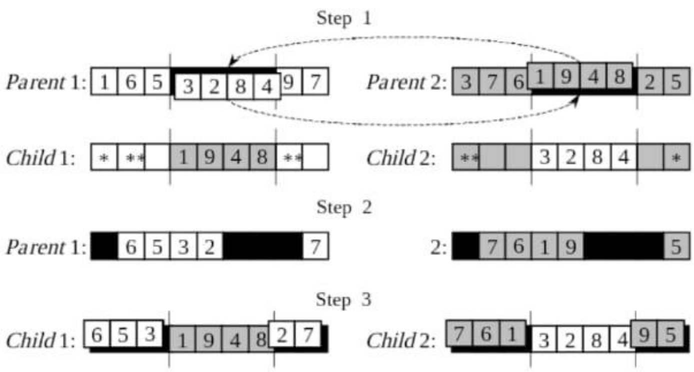


THE WORKING PROCESS OF THE GENETIC ALGORITHM :

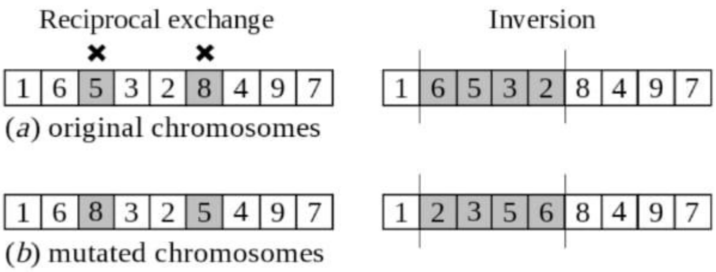
We need to decide how to represent the route of the salesman. The most natural way of representing a route is the path representation. Each city is given an alphabetic or numeric name, the route through the cities is represented as a chromosome,and appropriate genetic operators are used to create new routes.
Suppose we have nine cities named from 1 to 10 . In a chromosome, the order of the integers represents the order in which the cities will be visited by the salesman. So now let’s take a look at the picture given below.



Genetic Algorithm Working Model : *(Process)*



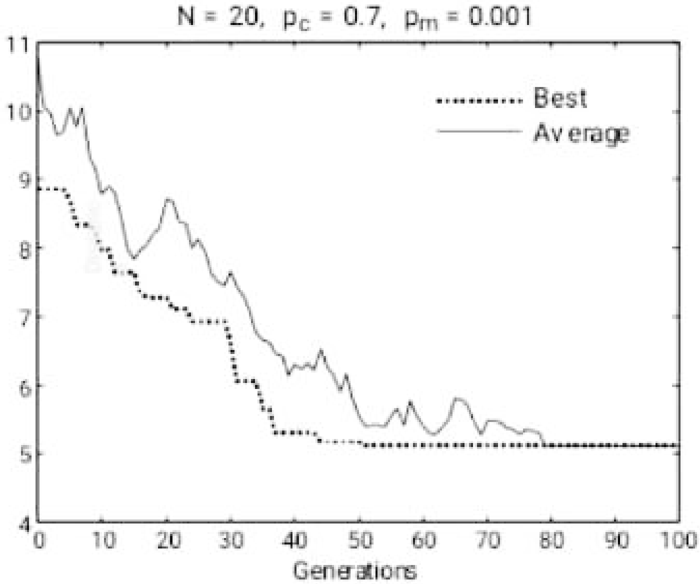
The **crossover** operator in its classical form cannot be directly applied to the TSP. A simple exchange of parts between two parents would produce illegal routes containing duplicates and omissions – some cities would be visited twice while some others would not be visited at all.

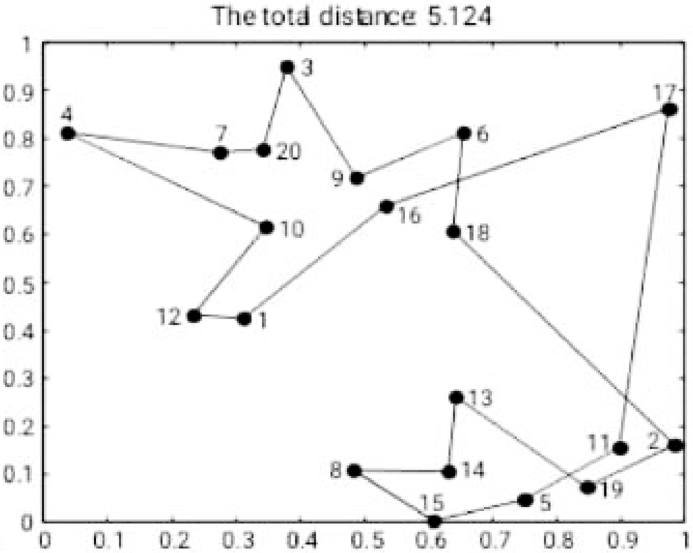


There are two types of **mutation operators**: Reciprocal exchange and Inversion. The reciprocal exchange operator simply   swaps two randomly selected cities in the chromosome.The inversion operator selects two random points along the chromosome string and reverses the order of the cities between these points.

Performance graph & the best solution route :

Total Generation | Mutation Rate | Population Config | Time





The Best Solution : (GA)

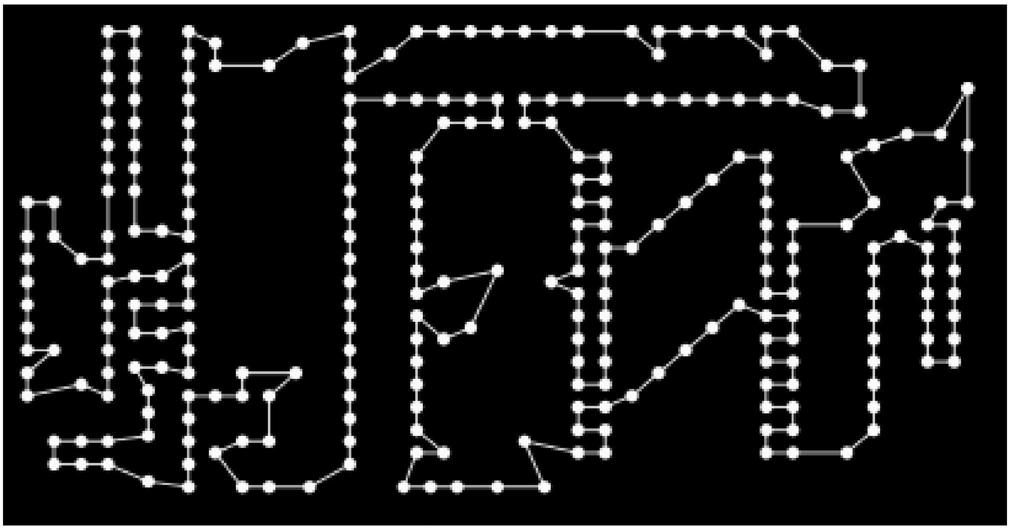


Fig :Genetic Algorithm (solving TSP)

Limitations of genetic algorithms:-

1) They are not effective in solving simple problems.
2) Lack of proper implementation may make the algorithm converge to a solution that is not optimal.
3) The quality of the final solution is not guaranteed.
4) Repetitive calculation of fitness values may cause some problems to experience computational challenges.

Conclusion :

In this literature we provide the basic overview of Genetic Algorithm as well as Travelling Salesman Problem.. The travelling salesman problem(TSP) is an optimization problem which is very simple at the point of understanding. It seems Like a very simple task but it is very difficult to solve. The problem is to find the shortest path through a set of points or cities or Whatever you may like to say and the only condition is each city is visited exactly once. The problem is known as NP-hard, and cannot be solved in polynomial time when the points are so large in number. Many algorithms have been developed in this case. Here we are going to use the Genetic Algorithm approach to solve this problem. And this is the main motive of this project.