A Solution of Genetic Algorithm for Solving Traveling Salesman Problem
Sonam Khattar¹ Dr. Puneet Gosawmi²
¹M.Tech Scholor ²Head & Professor
¹,² Department of CSE
¹,² GGITC, Ambala, India

Abstract— Genetic Algorithm is used to solve an optimization problems and Travelling Salesman Problem (TSP) is an optimization problem. TSP is an NP hard problem in combinational optimization important in operations research and theoretical computer science. The amount of computational time to solve this problem grows exponentially as the number of cities. These problems demand innovative solutions if they are to be solved within a reasonable amount of time. This paper explores the solution of Travelling Salesman Problem using genetic algorithms. The aim of this paper is to review how genetic algorithm applied to these problem and find an efficient solution.

Keyword:- TSP, Genetic Algorithm(GA), Selection, Mutation, Crossover.

I. INTRODUCTION
In the computer science, there is a field of artificial intelligence, in which genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called a Meta heuristic) is useful for generate a routinely solutions to optimization and search problems. Genetic algorithms is a kind of evolutionary algorithms (EA), which produce a solutions for an optimization problems using techniques such as selection, crossover, mutation, selection, and inheritance that are inspired by natural evolution [1]. John Holland proposed Genetic Algorithm in 1975 [2]. In the field of artificial intelligence a genetic algorithm is a search heuristic that mimics the process of natural evolution. Genetic Algorithm begin with various problem solution which are encoded into population the fitness of each individual is evaluating by applying a fitness function, after that with the process of selection, crossover and mutation a new generation is created.

An optimal solution for the problem is obtained, after the termination of genetic algorithm. If the condition for the termination of the genetic algorithm is not satisfied then algorithm continues with new population. The algorithm requires a population of individuals. Each individual is an encoded version of a proposed solution. The algorithm consists no of step beginning with the evaluation of individuals, selection of individuals, which will contribute to the next generation, recombination of the parents by means of number of operator such as crossover, mutation for the generation of new population. In this process, selection has the role of guiding the population towards some optimal solution, crossover the role of producing new combinations of partial solutions, and mutation the production of novel partial solutions. The genetic algorithm process consists of the following steps:[3]

A. [Start] Generate random population of n chromosomes (suitable solutions for the problem)
B. [Fitness] Find the fitness f(x) of each chromosome x in the population
C. [New population] Create a new population by repeating following steps until the new population is complete
D. [Selection] Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
E. [Crossover] is performed with a probability known as crossover probability that crossover the selected parents to produce a new offspring (children). If crossover probability is 0%, children are an exact copy of parents.
F. [Mutation] is performed with a probability known as mutation probability that decides which part of chromosome will be mutated. If mutation probability is 0%, nothing is changed
G. [Accepting] Place the newly generated offspring in a new population
H. [Replace] Use new generated offspring in the population for a further running of the algorithm
I. [Test] If the termination condition is satisfied, then stop the algorithm, and finally return to the solution that is best in current population
J. [Loop] Go to step 2

II. TRAVELLING SALESMAN PROBLEM
The Traveling Salesman Problem (TSP) is an optimization problem, which is solved very easily when the numbers of cities are less, but it is very difficult to solve, when the number of cities increases because very large amount of computation time is required. The numbers of approaches are used to solve the TSP. An approach genetic algorithm is used to solve TSP because of its robustness and flexibility. The problem is to find the shortest possible tour through a set of N vertices so that each vertex is visited exactly once. This Travelling Salesman Problem cannot be solved exactly in polynomial time and is known to be NP-hard.

Many exact and heuristic algorithms have been developed in the field of operations research (OR) to solve this problem. In the sections that follow, we are using genetic algorithm for solving TSP.

III. PROBLEM DEFINITION
I am developing here Genetic Algorithm (GA) for finding the solution of the Travelling Salesman Problem (TSP). The goal of a Traveling Salesman Problem, is to find out the
shortest distance between numbers of cities. The path taken by the salesman is known as a tour.

Testing of each and every possibility for a number of city tour would be N! Math additions. A 40 city tour would have to measure the total distance of be 8.16 X 10^{47} different tours. Adding one more city would cause the time to increase by a factor of 41. Obviously, this is an impossible solution.

IV. SOLUTION OF TSP USING GENETIC ALGORITHM

The solution of such type of problem can be finding using genetic algorithm in much less time. Sometimes the solution find is not the best solution; it can find the solution nearer to the best. There are number of steps for solving the traveling salesman problem using a GA.

1. Create a group of many random tours which is known as population. This algorithm uses a greedy initial population because it useful to give preference to linking cities those are near to each other.

2. Selection of the two parents i.e. two of the better (shorter) tours in the population and combines these selected parents to generate two (offspring) new child tours. With this hope, these children tour will be better than their parent.

3. The child tour is mutated after a small percentage of the time. This is done to prevent all tours in the population from looks like identical.

4. In the population new child tours are inserted and replacing the two of the longer tours by keeping the population size fixed.

5. The tour for the new children is repeated until the desired target is reached.

As the name implies, Genetic Algorithms mimic nature and evolution using the principles of Survival of the Fittest.

V. EXACT ALGORITHM

The exact algorithms are designed to find the optimal solution to the TSP, i.e., the tour of minimum length or distance. They are computationally expensive because they must (implicitly) consider all solutions in order to identify the optimum. These exact algorithms for the formulation of the TSP are typically derived from the integer linear programming (ILP)

Where the number of vertices is denoted by N, the distance between vertices i and j is denoted by dij and the are decision variables are denoted by xij's: xij is set to 0 or 1 when it is 1 when arc (ij) is included in the tour, and 0 otherwise. (xij) X denotes the set of sub tour-breaking constraints. The main reason for the study of this algorithm to propose a new method for the representation of chromosomes using binary matrix and new criteria for the fitness i.e to be used as method for finding the optimal solution for TSP.

\[ \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=1}^{n} dij xij \]

Subject to:

\[ \sum_{j=1}^{n} xij = 1, i=1...N \]

\[ \sum_{i=1}^{n} xij = 1, j=1...N \]

\[ xij = \{ 1 \text{ when arc (ij) included in tour} \}

\[ 0 \text{ otherwise} \]

That restricts the feasible solutions to those consisting of a single tour. Although the sub tour-breaking constraints can be formulated in many different ways without the sub tour breaking constraints, the reduction of the TSP to an assignment problem (AP), and a solution like the one shown in would then be feasible.

An optimal solution to the TSP can be find using Branch and bound algorithms, and the AP-relaxation is useful to generate good lower bounds on the optimal value. For particular asymmetric problems this is true, where dij for some i,j. For symmetric problems, like the Euclidean TSP (ETSP), the AP-solutions often contain many sub tours on two vertices. Consequently, these problems are better addressed by specialized algorithms that can exploit their particular structure. For instance, a specific ILP formulation can be derived for the symmetric problem which allows for relaxations that provide sharp lower bounds (e.g., the well-known one-tree relaxation.

VI. RESULT ANALYSIS

The travelling salesman problem using genetic Algorithm is coded in MATLAB. The performance of the algorithm was tested with a number of problems and an illustrative result is presented here in the fig.1.

![Locations of city](image)

Fig. 1: Locations of city

The determination of the distance matrix is the first step of proposed algorithm. The input data is given by locations of city as you see on Fig. 1 . It contains 20 locations. The task is to find out the optimal routes for these locations using the proposed algorithm. Figure 2 shows the distance matrix for the proposed algorithm.
A Solution of Genetic Algorithm for Solving Traveling Salesman Problem

Fig. 2: Distance Matrix
The first run shows the optimization solution to a distance of 3131.0134

Fig. 3: Result of the optimization in MATLAB
Results finds the minimum distances travelled by the Salesperson along with the number of iterations.

Fig. 4 Solution History

VII. CONCLUSION AND FUTURE WORK
In this paper, I have concluded how Genetic Algorithm can be used for solving the Travelling Salesman Problem. Genetic Algorithm finds the good solution for the Travelling Salesman Problem, depend upon the way how the problem is encoded and which types of crossover and mutation methods are used. A number of genetic algorithm techniques have been analyzed and surveyed for solving TSP. For the Researcher there is a lot of work to do in this field in future.

ACKNOWLEDGMENT
I would like to give my sincere gratitude to my guide Dr. Puneet Goswami who encouraged and guided me throughout this paper. Apart from his support as a professor and a researcher, I was fortunate to have his support as a friend.

REFERENCES