Volume 4, Issue 5, May 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Efficient Solution of Travelling Salesman Problem Using Genetic Algorithm

Sonam Khattar*
M.Tech Scholor,Department of CSE
GGITC,Ambala,India

Dr.Puneet Gosawmi
Head & Professor,Department of CSE
GGGI,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.

Keywords— TSP, Genetic Algorithm (GA), Selection, Mutation, Crossover.

I. INTRODUCTION

In the computer science field of artificial intelligence, genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called a Meta heuristic) is routinely used to generate useful 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 is a type of evolutionary algorithm. GA 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]

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
4. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
5. **[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.
6. **[Mutation]** is performed with a probability known as mutation probability that mutate new offspring at each locus (position in chromosome).
7. **[Accepting]** Place new offspring in a new population
8. **[Replace]** Use new generated population for a further run of algorithm
9. **[Test]** If the termination condition is satisfied, then stop the algorithm, and return the best solution in current population
10. **[Loop]** Go to step **2**

II. TRAVELLING SALESMAN PROBLEM

The Travelling 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 SPECIFICATION

The Travelling Salesman Problem (TSP) is a classic combinatorial optimization problem, which is simple to state but very difficult to solve. In travelling salesman problem, salesman travels n cities and returns to the starting city with the minimal cost, he is not allowed to cross the city more than once. In this problem we are taking the assumption that all the n cities are inter connected. The cost indicates the distance between two cities. To solve this problem we make use of genetic algorithm because the cities are randomly. Initial population for this problem is randomly selected cities. Fitness function is nothing but the minimum cost. Initially the fitness function is set to the maximum value and for each travel, the cost is calculated and compared with the fitness function. The new fitness value is assigned to the minimum cost. Initial population is randomly chosen and taken as the parent. For the next generation, the crossover is applied over the parent. This problem is known to be NP-hard, and cannot be solved exactly in polynomial time.

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 briefly introduce the OR problem-solving approaches to the TSP. Then, the genetic algorithms are discussed for the optimization and globalization of travelling salesman problem.

IV. TRAVELLING SALESMAN PROBLEM USING GENETIC ALGORITHMS

I am using here Genetic Algorithm (GA) for finding the solution of the Travelling Salesman Problem (TSP).The goal of a Travelling Salesman Problem, is to find the shortest distance between N different cities. The path taken by the salesman is called a **tour**. Testing every possibility for an N city tour would be N! Math additions. A 40 city tour would have to measure the total distance of be 8.16×10^{47} different tours. Adding one more city would cause the time to increase by a factor of 41.Obviously,this is an impossible solution.

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.

- First, create a group of many random tours in what is called a population. This algorithm uses a greedy initial population that gives preference to linking cities that are close to each other.
- Second, selection of the parents i.e. two of the better (shorter) tours in the population and combine them to make two (offspring) new child tours. With this hope, these children tour will be better than either parent.
- The child tour is mutated after a small percentage of the time. This is done to prevent all tours in the population from looking identical.
- In the population new child tours are inserted and replacing the two of the longer tours by keeping the population size fixed.
- The tour for the new children is repeated until the desired goal is reached.

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

V. PARAMETER OF THE GENETIC ALGORITHMS

There are 6 parameters used for controlling the operation of the Genetic Algorithm:

1. **Population Size** – When the algorithm starts the initial number of random tours that are created is known as population size. When the population size is larger it will takes longer time to find the result. A small size of population increases the chance in the population that every tour will eventually look the same. This increases the chance that the best solution will not be found.
2. **Neighborhood** - This number of tours are randomly chosen for each generation from the population. The two best tours become the parents. The worst two tours are replaced by the children. When the group size is high it increases the chance for good tours will be selected as a parent. A large group size will cause the algorithm to run faster, but it might not find the best solution.
3. **Mutation Percentage** - The percentage that each child after crossover will undergo **mutation** .When a tour is mutated, one of the cities is randomly moved from one point in the tour to another.
4. **#Nearby Cities** – As the genetic algorithm is a part of a greedy initial population, it will prefer to link cities that are nearer to each other to make the initial tours.
5. **Nearby City Odds percentage** - This is the percent chance that any one link in a random tour in the initial population will prefer to use a nearby city instead of a completely random city. If the GA chooses to use a nearby city, then there is an equally random chance that it will be any one of the cities from the previous parameter.
6. **Maximum Generations** - How many crossovers are run before the algorithm is terminated.

VI. PROPOSED ALGORITHMS

The 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 1when it is 1 when arc (i,j) is included in the tour, and 0 otherwise. (xij) X denotes the set of sub tour-breaking constraints. The main purpose of this study to propose a new representation method of chromosomes using binary matrix and new fittest criteria to be used as method for finding the optimal solution
for TSP.

$$\min \sum_{i=0}^{n} \sum_{j\neq i, j=1}^{n} dijxij$$

Subject to:

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

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

(xij) X

$$xij = \begin{cases} 1 & when\ arc\ (ij)\ included\ in\ tour \\ 0 & otherwise \end{cases}$$

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.[4]

VII. Implementation Of Proposed Algorithms

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.

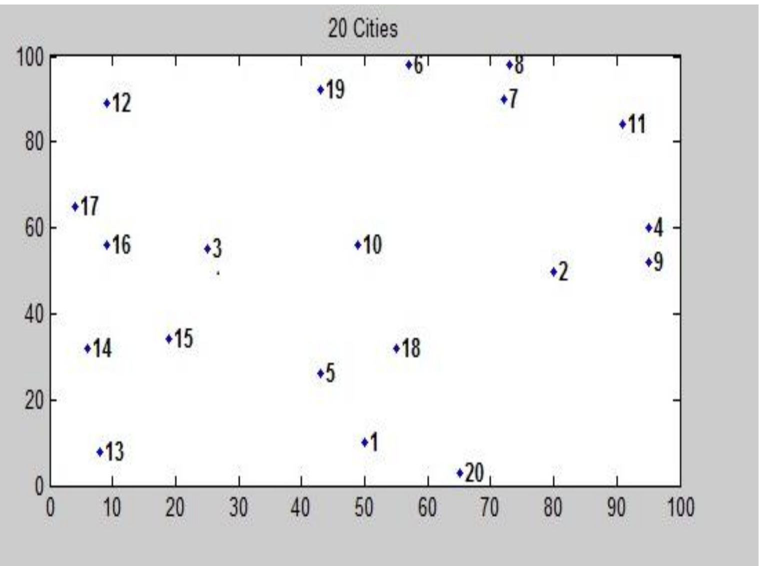

Fig. 1 Locations of city

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.

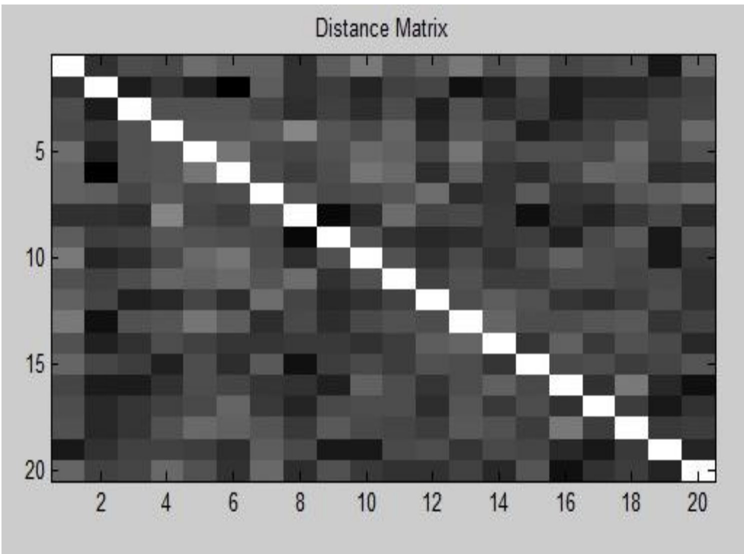

Fig. 2 Distance Matrix

The first run shows the optimization solution to a distance of 3131.0134

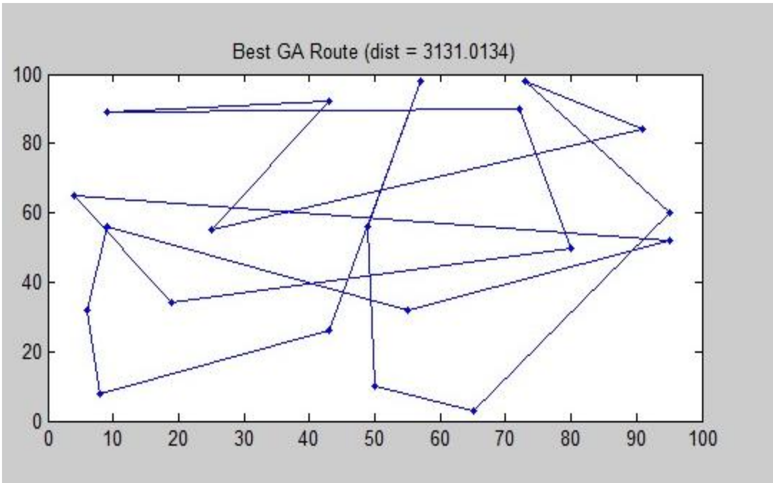

Fig. 3 Result of the optimization in MATLAB

Results finds the minimum distances travelled by the Salesperson along with the number of iterations

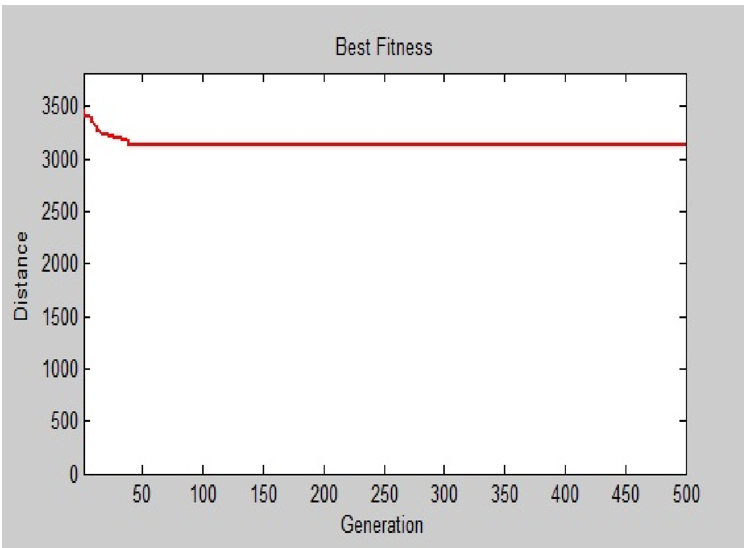

Fig. 4 Solution History

VIII. Conclusion And Future Work

In this paper, we see 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.

Acknowledgement

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

[1] http://en.wikipedia.org/wiki/Genetic_algorithm

[2] J. Holland, Adaptation in Natural and Artificial Systems : An Introductory Analysis with applications to biology , Control and Artificial Intelligence.‖ The University of Michigan Press,1975.

[3] http://www.obitko.com/tutorials/genetic-algorithms/ga-basic-description.php

[4] Jean-Yves Potvin , "Genetic Algorithms for the Traveling Salesman Problem" Centre de Recherche sur les Transports Université de Montréal C.P. 6128, Succ. A Montréal (Québec)
Canada H3C 3J7

[5]. Goldberg, Koza, Michalewicz and Beasley "Potvin ,Jean-Yves "new optimization technique for genetic algorithms" (Addison- Wesley, 1989)

[6] Monica Sehrawat, Sukhvir Singh,"Modified Order Crossover (OX) Operator " International Journal on Computer Science and Engineering (IJCSE) ISSN 0975-3397 Vol. 3 No. 5 May 2011

[7] Naveen kumar, Karambir and Rajiv Kumar, "A Genetic Algorithm Approach To Study Travelling Salesman Problem", Journal of Global Research in Computer Science, 2012, Vol. 3, No. (3).

[8] Saloni Gupta,Poonam Panwar "Solving Travelling Salesman Problem Using Genetic Algorithm" International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 6, June 2013

[9] Varunika Arya, Amit Goyal and Vibhuti Jaiswal "An Optimal Solution To Multiple Travelling Salesperson Problem Using Modified Genetic Algorithm" International Journal of Application or Innovation in Engineering & Management (IJAIEM) Volume 3, Issue 1,2014

[10] S.N.Sivanandam , S.N.Deepa "Introduction to Genetic Algorithms" Springer-Verlag Berlin Heidelberg 2008

