

**RESEARCH PAPER**

Available Online at [www.jgrcs.info](http://www.jgrcs.info)

## A STUDY OF GENETIC ALGORITHM TO SOLVE TRAVELLING SALESMAN PROBLEM

Naveen Kumar<sup>\*1</sup>, Karambir<sup>2</sup>, Rajiv Kumar

<sup>\*1</sup>Computer Science & Engineering Department University Institute of Engineering & Technology Kurukshetra University, Kurukshetra

<sup>2</sup>PhD Scholar CSE Deptt. Singhania University Jhunjhunu, Rajasthan,  
naveensaini85@gmail.com<sup>\*1</sup>

bidhankarambir@rediffmail.com<sup>2</sup>

rajiv.gill1@gmail.com<sup>3</sup>

**Abstract**— This paper presents the literature survey review of Travelling Salesman Problem (TSP). TSP belongs to the category of NP-hard problems. A various number of methods have been designed to solve this problem. Genetic Algorithm is one of the best methods which is used to solve various NP-hard problem such as TSP. The natural evolution process is always used by genetic Algorithm to solve the problems. This paper presents a critical survey to solve TSP problem using genetic algorithm methods that are proposed by researchers.

**Keywords**— TSP, NP- hard, Genetic Algorithm, Mutation, Selection, Crossover.

### INTRODUCTION

In operation research and computer science, the Travelling Salesman Problem (TSP) is an NP-hard problem. Using the branch and bound technique [1] and dynamic programming technique [2], TSP is solved very easily when there are less number of cities, but as the number of cities increases it is very hard to solve, as large amount of computation time is required. The number of fields where TSP can be used very effectively are military and traffic. In 1950s George Dantzig, Delbert Ray Fulkerson and Selmer M. Johnson expressed the problem as an integer linear program [3] and developed the cutting plane method for its solution. Another approach is to use genetic algorithm to solve TSP because of its robustness and flexibility. They are able to solve problems knowing nothing about of problem from the start [4].

### TSP PROBLEM

TSP is defined as: a set of  $n$  cities and their pair wise distance is given. A shortest possible path is to find out by traversing each city exactly once and returned to starting city from where salesman started. The salesman find a tour named Hamiltonian Circle must visit each and every city exactly once and returned to starting city and make the length or tour shortest or minimum. From graphical theory point of view it can be defined as, a graph  $G = (V, E)$  is given, a positive weight is assigned to each edge  $e \in E$ , the aim is to find the a Hamiltonian Circle  $C$  on graph  $G$  for which the sum of the weights is minimum[5].

$$W(C) = \min \sum_{e \in E(C)} w(e)$$

### GENETIC ALGORITHM

John Holland proposed Genetic Algorithm in 1975 [6]. In the field of artificial intelligence a genetic algorithm is a search heuristic that mimics the process of natural evolution. Genetic Algorithm belongs to class of evolutionary algorithm. GA begin with various problem solution which are encoded into population, a fitness function is applied for

evaluating the fitness of each individual, after that a new generation is created through the process of selection, crossover and mutation. After the termination of genetic algorithm, an optimal solution is obtained. If the termination condition is not satisfied then algorithm continues with new population.

The basic steps of genetic algorithm are given below:

#### **Initialization:**

An initial population is generated from many individual solutions. A problem depends upon size of the population that contains several hundreds or thousands of possible solutions. The search space contains all the possible solutions from which the population is generated randomly. However, the solutions are seeded in the areas from where the optimal solutions are likely to be found.

#### **Encoding:**

In Encoding, the genotype space is converted into phenotype space so that the genetic operations become more convenient. It converts the point in parameter space into bit string representation. For instance, a point (11, 6, 9) in three dimensional parameter space can be represented as a concatenated binary string as: 1011 0110 1001 in which each coordinate value is encoded as gene composed of four binary bits using binary coding. Encoding provides a way of translating problem specific knowledge directly into GA framework, thus play a key role in determining GA's performance.

#### **Fitness Function:**

Fitness function is usually converted from the objective function of a problem. It is used to evaluate the superior and inferior degree of a solution individual. If it has larger value then better solution is obtained. If it has smaller fitness value then better solution is not obtained. Another approach is to use the ranking of members in population as their fitness values.

#### **Selection:**

Selection operation simulates the natural law of the survival

of the fittest in the population evolutionary process. It determines the evolution direction which corresponds to the solution searching direction which corresponds to the solution direction in GA. In Selection operation, the individuals with larger fitness value are chosen to produce offspring, while the individuals with small fitness values are not chosen to produce offspring. Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover). Various selection techniques exist such as Roulette Wheel selection, Rank Selection, Steady state selection and Tournament selection.

**Crossover:**

Crossover operation is to simulate partially swap of information coming from chromosomes of two parent individuals. It symbolizes the biological heredity character. In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Crossover is a process of taking more than one parent solutions and producing a child solution from them. There are methods for selection of the chromosomes. Various crossover operator exists such as one point crossover, two point crossover, PMX, OX and Cyclic crossover.

**Mutation:**

Mutation operation is to simulate some genes on some offspring individual chromosomes can suddenly change into other genes with a very low probability in biological evolutionary process and produce new characters. It is helpful to generate new biological genetic pattern and search for the global optimal solution. For TSP problem mutation operations have several methods including insert Mutation, Swap Mutation, Inversion Mutation, and Scramble Mutation. In genetic algorithms of computing, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set to high, the search will turn into a primitive random search.

**Termination:**

When the algorithm has run a given number of iterations, it stops and output the best solution. This generational process is repeated until a termination condition has been reached.

**SURVEY REPORT OF TSP**

In 1999 P. Larranaga (et al)[18] explained various crossover and mutation operators that were developed to solve TSP. These are shown in table 3.1.

Table: 1

Representation	Operators	Authors
Binary	Classical + repair operator	Lidd (1991)

Path	Partially mapped crossover	Goldberg and Lingle (1985)
	Order crossover	Davis (1985)
	Order based crossover	Syswerda (1991)
	Position based crossover	Syswerda (1991)
	Heuristic crossover	Grefenstette (1987b)
	Edge recombination crossover	Whitley et al. (1989)
	Sorted match crossover	Brady (1985)
	Maximal preservative crossover	M'uhlenbein et al. (1988)
	Voting recombination crossover	M'uhlenbein (1989)
	Alternating positions crossover	Larrañaga et al. (1996a)
	Displacement mutation	Michalewicz (1992)
	Exchange mutation	Banzhaf (1990)
	Insertion mutation	Fogel (1988)
	Simple inversion mutation	Holland (1975)
	Inversion mutation	Fogel (1990)
Scramble mutation	Syswerda (1991)	
Adjacency	Alternating edge crossover	Grefenstette et al. (1985)
	Sub tour chunks crossover	Grefenstette et al. (1985)
	Heuristic crossover 1	Grefenstette et al. (1985)
	Heuristic crossover 2	Jog et al. (1989)
	Heuristic crossover 3	Suh and Van Gucht (1987)
Ordinal	Classical operators	Grefenstette et al. (1985)
Matrix	Intersection crossover operator	Fox and McMahon (1987)
	Union crossover operator	Fox and McMahon (1987)
	Repair operators	Seniw (1981)
	Repair operators	Homaifar and Guan (1991)
	Heuristic inversion mutation	Homaifar and Guan (1991)
Permutation	Modified Maximal Preservative Right Side Placement Crossover(MMPX)	Rajiv et.al. (2012a)
	SORT + PMX with Same Position Crossover(SPMX-1)	Rajiv et.al. (2012b)

In 2003 Peng Gang (et al) [7] described the method to solve the Travelling Salesman Problem using a multiple heuristic search algorithm. Two main operations of Complete 2-Opt (C2Opt) method and smallest square (SS) method were combined in a genetic algorithm (GA) and applied to TSP. The C2Opt was based on the 2-Opt heuristic search which removed all crossed edges in the tour. The SS selected the shorter edges than C2Opt. However a reasonable result was presented by combining the C2Opt and SS in GA for TSP. Another two operations the deletions (DL) and the Best Part Collector (BPC) were discussed. The DL was used for the removing the duplicates from population and the BPC was used for collecting the best part among the individuals to elite individual.

To solve the TSP, there were currently three general classes of heuristics: classical tour construction heuristics such as Nearest Neighbour method, the Greedy algorithm and Local

Search algorithms based on rearranging segments of tour. The 2-Opt heuristic search had been used to optimized TSP tours in connection with genetic search. Usually the 2-Opt was randomly implemented by the GA to 20 – 30 % of the individuals of the population. The 2-Opt removed the two edges in a tour and then one of the result segment was reserved and two segments were reconnected. If 2-Opt results in an improved tour, the changes were kept. Otherwise the tour was returned to original form. The 2-Opt was typically applied to all  $(n^2 - n) / 2$  pair of edges.

In 2004 W-ping wang (*et al*) [8] described that traditional crossover operator perform the constant merging between two Parents without considering their homogeneity. In their work, a self-adaptive mechanism named Adaptive Recombination 'with Three Sub-populations (ARTS) was therefore presented to control the crossover operator of a genetic algorithm. The ARTS allowed the crossover to be dynamically switched among two-point crossover (i.e, the least disruptive crossover), and uniform crossover with probability 0.5 (i.e., the most disruptive crossover). Their work has presented a self-adaptive mechanism named the Adaptive Recombination with Three Sub-populations (ARTS) to control the crossover operator of a genetic algorithm.

Again in 2004 Huai-Kuang Tsa (*et al*) [9] proposed an evolutionary algorithm, called the heterogeneous selection evolutionary algorithm (HeSEA), for solving large travelling salesman problems (TSP). The strengths and limitations of numerous well-known genetic operators were first analyzed, along with local search methods for TSPs from their solution qualities and mechanisms for preserving and adding edge Based on that analysis, a new approach, HeSEA was proposed which integrated edge assembly crossover (EAX) and Lin-Kernighan (LK) local search, through family competition and heterogeneous pairing selection. The study demonstrated experimentally that EAX and LK could compensate for each other's disadvantages. Family competition and heterogeneous pairing selections were used to maintain the diversity of the population, which was especially useful for evolutionary algorithms in solving large TSPs. The proposed method was evaluated on 16 well-known TSPs in which the numbers of cities range from 318 to 13509. Experimental results indicated that HeSEA performed well and was very competitive with other approaches. The proposed method determined the optimum path when the number of cities was under 10,000 and the mean solution quality was within 0.0074% above the optimum for each test problem.

In 2007 Yingzi Wei Yulan Hu and Kanfeng Gu [10] described that traditional genetic algorithms often met the occurrence of slow convergence and enclosure competition. They presented a parallel algorithm for the travelling salesman problem (TSP) which incorporated several greedy heuristics based on genetic algorithms. This algorithm was called the Greedy Genetic Algorithm (GGA). The ideas they incorporated in their algorithm included : (i) generating the initial population using the gene bank (ii) double-directional greedy crossover and the local searches of greedy mutation (iii) special-purpose immigrations to promote diversity of population and an open competition (iv) stepwise parallel

optimization of each individual of the population (v) developing an overall design that attempt to strike a right balance between diversification and a bias towards fitter individuals. They tested all these ideas to assess their impact on the GGA performance, and also compared their final algorithm to the simple genetic algorithm. They found the Greedy Genetic Algorithm to be a very effective and robust algorithm for TSPs.

In encoding scheme a path representation was used where the cities were listed in the order in which they were visited. In this technique the N cities were represented by a permutation of the integers from 1 to N. For example, assuming there were 5 cities 1, 2, 3, 4 and 5, if a salesman went from city 4, through city 1, city 2, city 5, city 3 and returned back to city 4, the chromosome was {4 1 2 5 3}. For an N cities TSP, the population was initialized by randomly placing 1 to N length chromosomes and guaranteed that each city appearing exactly once. Thus chromosomes stand for legal tour.

In initial population generation the N cities were permuted and assembled to build a gene bank. For a TSP of N-cities, C cities that were closer to cities i were encoded to construct a gene bank where C was a number less than N-1. For simplifications, C equal 3 in GGA. Gene bank was a matrix  $A_{N \times C}$  whose size was  $N * C$ . The element of  $A[i] [j]$  was the closest city to city i. For example  $A[i] [1]$  and  $A[i] [2]$  were first and second city closest to city i respectively.

The Double Directional crossover was same as that of greedy crossover invented by Grefenstette. Greedy crossover select the first city of one parent, compared the cities leaving that city in both parents and choosed the closer one to extend the tour. If one city has already appeared in tour, then other city was chosen. In greedy mutation operator, greedy swap of two cities positions. The basic idea of greedy swap was to randomly select two adjacent cities from one chromosome and swap them the new tour length was shorter than the elder. GGA keep the new tour only when getting a shorter length tour after not more than 3 trials of swap. The greedy mutation operation was a procedure of local adjustments and improvement for the chromosomes.

In 2009 Omar M.Sallabi and Younis El-Haddad [11] described the Improved Genetic Algorithm (IGA) to solve Travelling Salesman Problem. In that, new crossover operation, population reformulates operation, multi mutation operation, partial local optimal mutation operation and rearrangement operations were used to solve the TSP problem. The proposed IGA was compared with three GA's which used different crossover operations and mutations.

The IGA was concerned with his operation more than population size. Thus the initial population only consisted of two individuals. Applying population reformulated the operation. A Swapped Inverted Crossover (SIC) was applied to these individuals to produce 12 children with different characteristics inherited from their parent. Ten copies were made of these children by applying multi mutation where each copy mutate with different method. The fitness function for each individual was evaluated and the best two selected. Finally a partial local optimal mutation operation

for next generation was applied. The technique used for IGA was : the tour was divided into three parts, with two cut points ( $p_1$  and  $p_2$ ), the head containing  $(1, 2, 3, \dots, p_1-1)$ , the middle containing  $(p_1, p_1+1, \dots, p_2)$ , and the tail containing  $(p_2+1, p_2+2, \dots, n)$ . The SIC changing the head of first parent with tail of second parent. The middle remains unchanged until the partial local optimal mutation operation was applied.

This improved the middle tour by finding its local minima. The role of population reformulated operation is to change the structure of the tour by exchanging the head and tail with middle. Swapped Inverted Crossover (SIC) was used to improve the GA performance. The main idea was to backtrack different ways to search for better tours. That crossover was applied with a one or two point crossover. In Two point SIC, the basic principle was two random cut points ( $p_1$  and  $p_2$ ), a head containing  $(1, 2, \dots, p_1-1)$ , the middle containing  $(p_1, p_1+1, \dots, p_2)$  and tail containing  $(p_2+1, p_2+2, \dots, n)$ . The head and tail of each parent were flipped and then the head of first parent was swapped with tail of other parent.

While in One Point SIC, the selecting one crossover point randomly, the head of parent1 was flipped and swapped once with the head of parent 2 and second with tail of parent 2.

Again in 2009 Gohar Vahdati (et al) [12] proposed a new solution for Travelling Salesman Problem (TSP), using genetic algorithm. A heuristic crossover and mutation operation was proposed to prevent premature convergence. Presented operations tried not only to solve that challenge by means of a heuristic function but also considerably accelerated the speed of convergence by reducing excessively the number of generations. By considering TSP's evaluation function, as a travelled route among all  $n$  cities, the probability of crossover and mutation have been adaptively and nonlinearly tuned. Experimental results demonstrated that proposed algorithm due to the heuristic performance was not easily getting stuck in local optima and has a reasonable convergent speed to reach the global optimal solution.

To generate and to reinforce optimal chromosomes, heuristic crossover and mutation operations have been suggested. Probability of these operations has been tuned adaptively and nonlinearly to avoid premature and slow convergence simultaneously, as well as low stability. Here, a heuristic crossover operator, which increased the algorithm speed, has been proposed. Although it was mentioned that due to the goal function of TSP which was finding the shortest total distance in one closed cycle parent chromosomes represented in a closed cycle. They tuned probabilities of crossover and mutation operations adaptively and nonlinearly. Tuning curves of crossover and mutation changed slowly. One of the permanent challenges for GAs was how to deal with premature convergence due to a sudden and fast reduction of search space and also getting stuck in local optima. Here a GA based on heuristic crossover and mutation was proposed to solve the travelling salesman problem. Suggested heuristic crossover, actually used four pointers, one pair for each parent, which moved

clockwise and counter clock wise. These pointers were evaluated by means of a fitness function, equal to the inverse distance between two cities, and try not to get stuck in local optima. That approach improved the convergent speed towards the global optimal solution. The heuristic mutation prevented premature convergence too. The role of adaptive and nonlinear probabilities of crossover and mutation was used to solve the low stability and also slow convergent.

Again in 2009 Fanchen Su (et al) [13] described a novel crossover operator, Cut-blend crossover, for a genetic algorithm for the TSP. Cut-blend crossover was the best in such kind of crossovers that improved a tour using a sub-tour extracted from other tour or tours (PMX, OX). The proposed operators were embedded in a new genetic algorithm, which extracted sub-tours from a pool consisting of former best tours and current population, to compare with other crossovers. The operator was evaluated on a number of well-known benchmark. Experimental results showed that the new crossover was superior to the conventional crossovers such as OX, ER, especially in problems of larger scale role of crossovers was to generate offspring that were better tours by preserving partial tours from the parents. In order to make full use of the former tours, a set is constructed based on former best tours of every generation, and current population. They named the set Eugenic Tour Pool. A module was extracted from Eugenic Tour Pool, which was used for Cut-blend operator

In this the edges with high ranks were included in the optimal tour had the higher probability than those with low ranks. When a new edge was to be added in crossover operator, edges with high ranks were preferred. Eugenic Tour Pool was made up of part of former best tours (best tours of every generation) and the current population. Cut-blend crossover was to add the module extracted from other tour or tours into the tour. In order to insert the module, some edges were deleted, and then the current tour was divided to some sub tours. This operator was noted as Cut Operator. The sub tours produced by Cut Operator linked together to construct a complete tour, and this process was completed by the Blend Operator. A new crossover operator, Cut-blend crossover, and a genetic algorithm based on Cut-blend crossover were developed. Experiments showed that Cut-blend crossover outperformed other conventional crossovers, such as OX, PMX.

In 2010 Chunxiang Wang and Xiaoni Guo [14] described a hybrid algorithm (HA) integrated genetic algorithm (GA) with ant optimization (ACO) to solve TSP to get better optimization performance than each single algorithm and the complemented the advantages of each other and avoided each own limitations. This algorithm run GA first and after that ACO. A new strategy called GSA was proposed aiming at the key link in the HA that converted the genetic solution from GA into information pheromone to distribute in ACO.

GAs took the new matrix which was formed by the combination of former 90% of individual from genetic solution and 10% of individual by random generation as the basis of transformation of pheromone value. The best combination of genetic operators in GA was also discussed.

Several TSP were used as simulation test to test genetic operators matching and optimization performance of HA.

The result showed that PMX crossover matched with IVM mutation in GA proved to be best combination of genetic operators which was able to make the GA improve the precision of optimal solution and HA using the best combination operators and GSA strategy was successful and available to search for optimal solution in high efficiency and good convergence.

In 2011 Luo Delin (*et al*) [15] described a Heuristic Simulated Annealing Genetic Algorithm (HSAGA) to solve TSP problem in which Genetic Algorithm (GA) was used as global search and the Heuristic Simulated Annealing (HSA) Algorithm was used as local search. This algorithm was applied on partial optimal solutions at each iteration. The function HSA was used to enhance the search effectiveness over the solution space and to avoid getting stuck into the local optimal trap. The author described the local search strategy with SA (Simulated Annealing). SA was an optimization algorithm which simulated the atomic crystallization process when heated metal cool down slowly and anneal. Besides accepting better solution, SA also accepted inferior solution with certain probability thus it escaped from local optima and find the global optimal solution. Since pure GA has slow convergence rate and its search process was coarse. Here SA was imported as a local search mechanism and was applied to a certain proportion of optimal solutions obtained at each running of GA. Here the goal of introducing SA was to intensively search the neighbourhood of a solution for better solutions and helped avoid getting stuck into local optima.

Rajiv et al. (2012,b)[16], rajiv et. al. (2012,c)[17] have proposed the new crossover operator for the OSPSP. MMPX, SPMX-1 operators are permutation encoding based operator. It was observed from the literature survey that these operators will be very suitable for the TSP problem.

## CONCLUSION

A number of genetic algorithm techniques have been analysed and surveyed for solving TSP. The researchers have been used a number of genetic operators such as crossover; mutation and their combination to find the solution of TSP are studied here. After analysing the literature review of TSP, it is observed that there is requirement to design new genetic operators that can enhance the performance of the GA used to solve TSP. There is lot of scope for the researcher to do work in this field in future.

## REFERENCES

- [1]. S. Tschoke, M. Racke, R. Luling, B. Monien, "Solving the Travelling Salesman Problem with Parallel branch and bound algorithm on 1024
- [2]. Jellouli olfa , Chatelet eric, " Dynamic Programming approach for generalized travelling salesman problem",2000.
- [3]. [http://knowledgerush.com/kr/encyclopedia/travelling\\_salesman\\_problem/](http://knowledgerush.com/kr/encyclopedia/travelling_salesman_problem/)
- [4]. John R koza,"Survey of genetic algorithms and genetic programming",2009.
- [5]. Y.P.Wang, Y. H. Li," A novel quantum genetic algorithm for TSP",Chinese journal of computers, 30(2):pp748-755,2009.
- [6]. 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.
- [7]. Peng Gang et al, "Multiple Heuristic Search Algorithm for Solving Travelling Salesman Problem", IEEE, 2003.
- [8]. W-ping wang et al, "A dynamically switched crossover for genetic algorithms", IEEE, 16 January 2004, Volume 168, Issue 2, pp 390-402.
- [9]. Huai-Kuang Tsai et al, "An Evolutionary algorithm for Large Travelling Salesperson Problems", IEEE, February-May 2004, Volume 11, Issues 1-2, pp 113-127.
- [10]. Yingzi Wei Yulan Hu & Kanfeng Gu, "Parallel Search Strategies for TSP Using a Greedy Genetic Algorithm", IEEE, 2007.
- [11]. Omar M.Sallabi , Younis El- Haddad, "An improved Genetic Algorithm to solve Travelling Salesman Problem", IEEE, 2009, pp 471-474.
- [12]. Gohar Vahdati et al, "A new approach to solve TSP using Genetic Algorithm based on Heuristic Crossover and Mutation operator", IEEE, 2009, pp 112-115.
- [13]. Fanchen Su et al, "New Crossover Operator of Genetic Algorithms for the TSP", International Joint Conference on Computational Science and Optimization, IEEE 2009, pp 666-669.
- [14]. Chunxiang Wang & Xiaoni Guo, "A Hybrid algorithm based on genetic algorithm and Ant Colony Optimization for TSP", IEEE, 2010.
- [15]. Luo Delin, Zhang Lixiao & Xu Zhihui, "Heuristic Simulated Annealing genetic algorithm for TSP", IEEE, 2011, pp 260-264.
- [16]. Kumar,R., Joshi, M.M, Gill, S.,Ray, H.M.. (2012c) " An Experimental Approach to Analyze the Different Types of Stopping Criteria used in Genetic Algorithm under (OSPSP) , National Level Seminar on Applications of Mathematics in Engineering Sciences,(AMES2012)
- [17]. Kumar,R., Joshi, M.M, Gill, S.,Ray, H.M.. (2012c) "A Simulation Based Study of Genetic Algorithm under Operating System Process Scheduling Problem (OSPSP) " , National Level Seminar on Applications of Mathematics in Engineering Sciences,(AMES2012)
- [18]. Larranga, P et al "Genetic Algorithm for Travelling Salesman Problem :A review of representation and operators", Artificial Intelligence Review13,1999, pp 129-170