Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.677 – 687

RESEARCH ARTICLE

Different Genetic Operator Based Analysis and Exploration of TSP

Ashima Malik

Computer Science Department, Banasthali Vidyapith, India

ashima.13malik@gmail.com

Abstract: TSP has been considered the most complex algorithmic problem because of its time complexity. TSP comes under the NP Complete problem that becomes more critical as the number of cities increases. In this present work, an effective solution to TSP is provided using genetic approach. The work has presented genetic based model to generate the TSP path in effective time. The improvement is here performed on fitness function and crossover stages where the cost based analysis is performed to generate the effective path. In this paper, a detailed description to genetic process model is given with exploration of different stages of genetics. The paper also includes the presented algorithm along with associated assumptions. The work will be able to provide effective solution in optimized time.

Keywords—Travelling Salesman Problem; Genetics; Crossover; Mutation; Selection; Reproduction

I. INTRODUCTION

TSP: TSP can be defined as a problem in which we are given a complete undirected graph say G=(V,E) that has non-negative integer cost c(u,v) belongs to E, and we must find a Hamiltonian cycle(a tour) of G with minimum cost. In other words it can be defined as a most famous problem of dynamic programming in which a salesman has to visit 'n' cities defined in form of network represented as complete graph, where salesman has to visit all cities only once with minimum distance and return back to starting city.[6]

A. *Requirements of tsp*:

The main aim of tsp is to search the shortest or we can say the cheapest tour for a salesman to visit all cities exactly once and return back to starting city.[1] This problem comes under Artificial Intelligence and defines some sub-problems like shortest path, Hamiltonian Cycle problem etc. Here solution is found in a definite and optimized time.

B. Applications of tsp:

- Vehicle routing problem is an example of Traveling Salesman Problem where the problem is to find which customers should be served by which vehicles and the minimum number of vehicles needed to serve each customer. There are different variations of this problem including finding the minimum time to serve all customers. Some of these problems can be modeled as the TSP[11].
- The problem of computer wiring can also be modeled as a TSP, where number of pins represents several modules. Subsets of these pins are connected with wires in such a way that no pin has more than two wires attached to it and the length of the wire is minimized.
- Another application is overhauling gas turbine engines in aircraft. Nozzle-guide vane assemblies, consisting of nozzle guide vanes fixed to the circumference, are located at each turbine stage to ensure uniform gas flow. The placement of the vanes in order to minimize fuel consumption can be modeled as a symmetric TSP [14].
- The scheduling of jobs on a single machine given the time it takes for each job and the time it takes to prepare the machine for each job is also TSP. Objective is to minimize the total time to process each job[15].
- TSP is also used in the field of robotics. A robot must perform many different operations to complete a process. In this application, as opposed to the scheduling of jobs on a machine, there are precedence constraints. This is an example of a problem that cannot be modeled by a TSP but methods used to solve the TSP may be adapted to solve this problem [3][4].

C. Variations of tsp:

There are many different variations of the Traveling Salesman Problem:

- Shortest Hamiltonian Path Problem**:** If in a graph, each edge has a weight and two nodes Vs and Vt are given and objective is to find the shortest Hamiltonian path from Vs to Vt. If an edge from Vt to Vs is added and give it weight M where M is large and positive, then optimal TSP tour will always include this edge (because it will reduce the cost of the tour) and will therefore solve the Hamiltonian problem[8].
- The Asymmetric Traveling Salesman Problem**:** When the cost of traveling from city i to city j is not the same as the cost from city j to city i then it is called Asymmetric Traveling Salesman Problem. This can be solved in the same way as the standard TSP if certain edge weights are applied that ensure that there is a Hamiltonian cycle in the graph[10].
- The Multi-salesmen Problem**:** It is the same as the standard TSP except that there is more than one salesman. Problem is to decide where to send each salesman so that every city is visited exactly once and each salesman returns to the original city[11].
- The Bottleneck Traveling Salesman Problem: In this type of TSP, aim is to minimize the largest edge cost in the tour instead of the total cost. That is, to minimize the maximum distance the salesman travels between any two adjacent cities[14].
- The Time Dependent Traveling Salesman Problem**:** It is the same as the Standard Traveling Salesman Problem except time periods are also included. The cost C_{ijt} is the cost of traveling from node i to node j in time period t. [15].

D. Complexity of tsp**:**

Tsp is a NP-Complete problem, whose status is unknown means no polynomial time algorithm has yet been discovered for them, nor has anyone yet been able to prove that no polynomial time algorithm can exist for anyone of them. Several of NP-complete problems are similar to problems that have polynomial time algorithms. In each of the following pair of problems, one is solvable in polynomial time and other is NP-complete. Following are the variations for NP-complete problems:

- Shortest vs. longest simple paths: In a directed graph G=(V,E) shortest path could be found from a single source even with negative edge weights in O(VE) time.
- Euler tour vs. Hamiltonian cycle: We can determine whether a graph has an Euler tour in O(E) time but determining the existence of Hamiltonian cycle in a graph is NP-complete problem.
- 2-CNF satisfiability vs. 3-CNF satisfiability: There exists a polynomial time algorithm to determine satisfiability of 2-CNF formula but the problem to determine satisfiability of 3-CNF is NP-complete.[2]

E. Evolutionary Algorithm:

An evolutionary algorithm can be defined as a heuristic optimization algorithm that uses techniques which are inspired by mechanics from organic evolution such as mutation, recombination, and natural selection to find an optimal configuration for a specific system within specific constraints. In terms of computer science evolution strategy can be defined as an optimization technique based on ideas of adaptation and evolution methodologies.[9]

II. GENETICS

Genetic algorithms are modeled on biological processes in which parents pass character traits to their offspring. The next generation contains data inherited from its predecessors and in each generation the fittest members have the greatest potential to survive and send genetic material to the progeny of their population. As children are developed from the best parents, they are likely to introduce an improvement in fitness to the group.[9]

Genetic algorithms mimic this survival of the fittest by randomly generating a population of solutions and then selecting members, with greater possibility of selection given to the fittest, from which to build the next generation [13]. An example explained here is of study that used populations of 100 and evolved each trial for 1000 generations. The objective function used in this research was the length of the paths. Shorter circuits were given best fitness consideration by inverting their tour lengths through subtraction from the ceiling of the population's longest tour [11]. The roulette wheel method was employed to choose parent solutions.

Successors were developed by binary operations called crossovers that create child solutions using information inherent in the two chosen parents. The type of information passed is problem dependent and affects the fitness of the resultant population. Unary operators purpose is to allow solutions to exceed local maximums but they are usually destructive to the mutated offspring.[6]

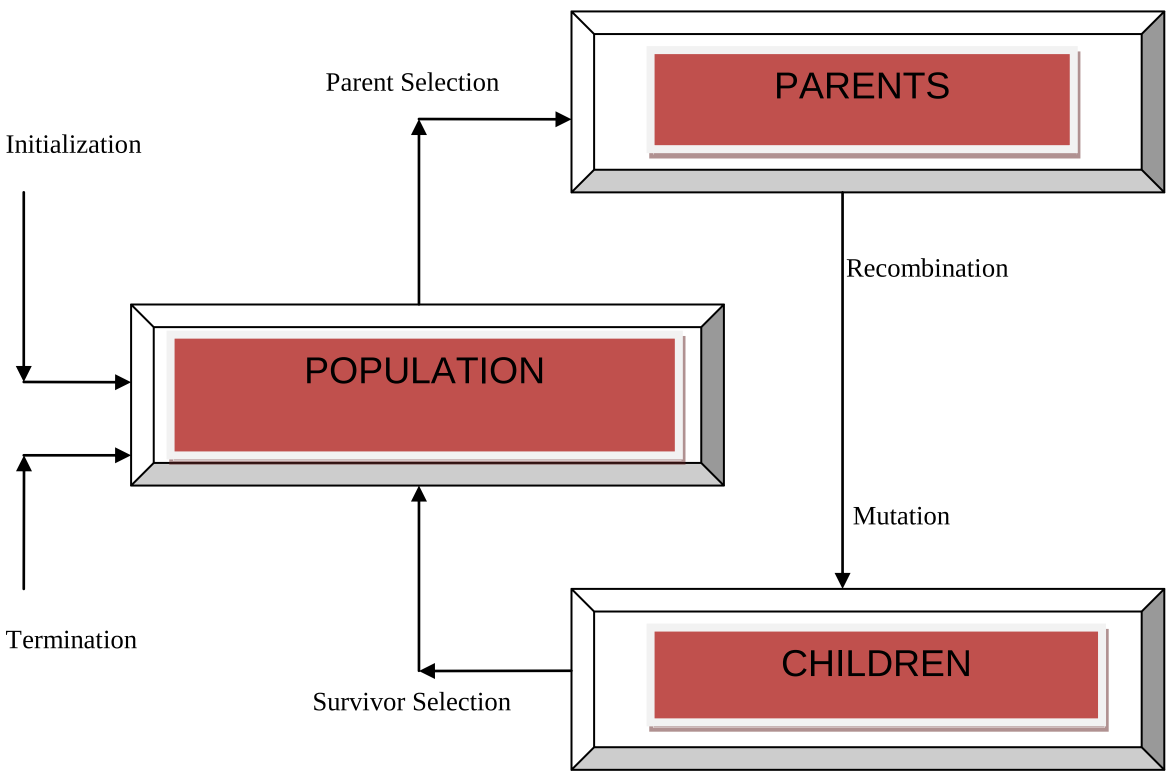

Fig 1 showing Basic Genetic Model

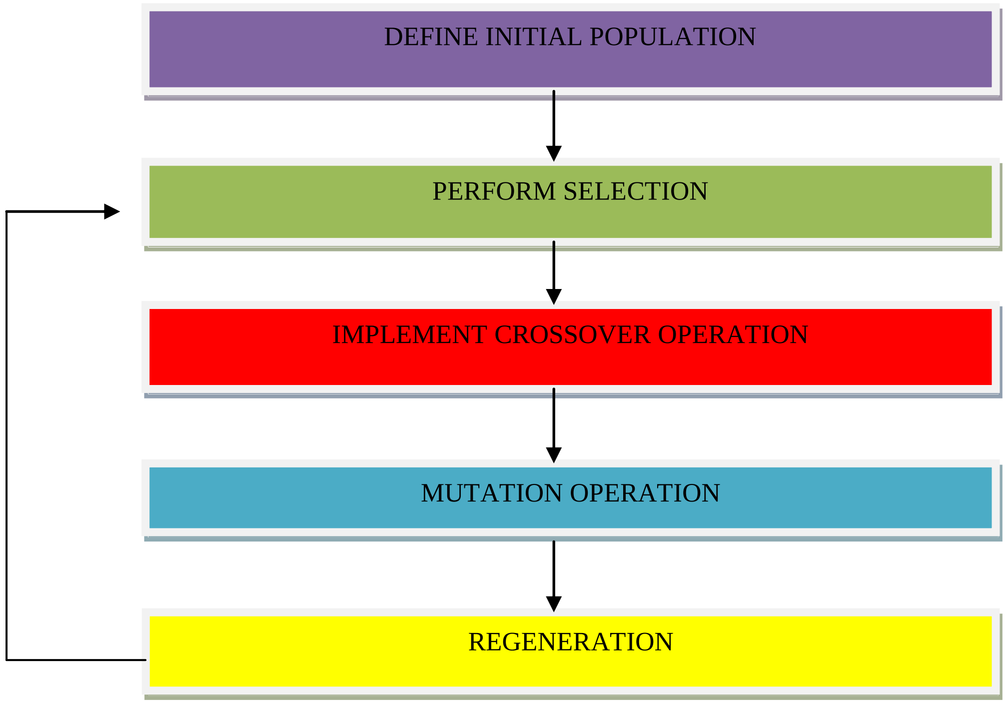

Fig 2 Showing Flow of Work

- Defining initial population: Initial population is chosen at random through successive flips of a unbiased coin. Then a simple set of operations are defined that take this initial population and generate successive populations.
- Perform Selection: Selection is how to choose the individuals in the population that will create offsprings from the next generation and how many offsprings each will create.[6] The purpose of selection is to emphasize the fitter individuals in the population in hopes that their offsprings will in turn have even higher fitness. Selection can be too-strong or too-weak but it should be balanced with variation from crossover and mutation. In too-strong selection the sub-optimal highly fit individuals will take over the population, reducing the diversity needed for further change and progress. Too-weak selection results in too-slow evolution.[3].

A. *Various selection methods*:

- Roulette wheel selection
- Boltzman selection
- Tournament selection
- Rank selection
- Steady-state selection

- Roulette wheel selection:

This method is used to select the particular candidate solution. Here probability of solution is proportional to the fitness. Here proportionate reproductive operator is used which selects a string from the mating pool with a probability proportional to fitness.[10] The sum of probabilities of each string being selected from mating pool must be one. The probability if ith selected string is:

$$\bar{F}=\sum_{j=1}^{n} Fj/n$$

Expression showing that ith string in the population is selected with a probability proportional to fi where fi is the fitness value for that string and 'n' is the population size.

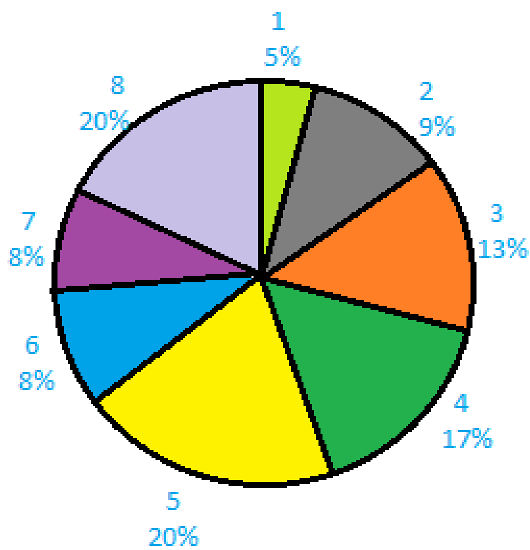

Fig3 Showing Roulette wheel marked for eight individuals according to fitness

- Boltzman selection method**:**

According to this method if temperature is high then there will be equal chances of selection of very chromosome but when temperature will decrease, the probability of selection of highly fitted chromosomes will be high and less fitted chromosomes will be low.

$$P(E)= \exp(\frac{-E}{kT})$$

Where 'k' is Boltzman constant. This expression suggests that a system at a high temperature has almost uniform probability of being at energy state, but at a low temperature it has a small probability of being at a high energy state. Therefore convergence of algorithm is controlled by temperature T and Boltzman probability distribution.[7]

- Tournament selection**:**

This method can be explained in following steps:

1. Two individuals are selected Nu=2,
2. A random number 'r' is generated between 0 and 1,
3. Then a constant 'k' is taken as K=0.75,
4. If generated random number 'r' is less than 'k' i.e. r<k then best fitted individual will be selected as parent. But if r>k, the least fitted will be selected as parent. (Again these two are added to the population),
5. Process will be repeated n times[12].

- Rank selection**:**

It came to overcome the limitation of fitness proportionate method. Here ranking of individuals is done according to fitness after sorting them. Subjective fitness is considered herein place of absolute fitness. In other words rank selection is a method that first ranks the population and receives fitness from the ranking taking every chromosome. The worst will have fitness 1, next will have 2 ….and the best will have fitness N.

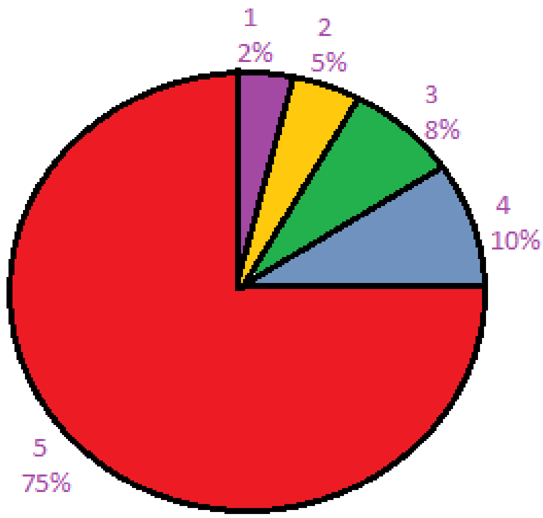

Fig4 Showing Roulette wheel according to fitness

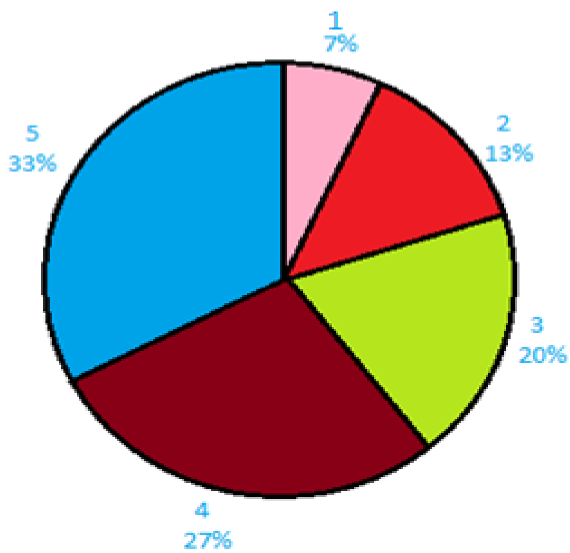

Fig5 Showing Roulette wheel according to Rank

- Steady-state selection**:**

In this method Genetic algorithm works in the following way. In every generation a few chromosomes with high fitness are selected (for maximization problem) for creating new offsprings. Then some bad chromosomes i.e. with low fitness are removed and new offspring is placed in that place. The rest of population survives a new generation.[15]

- Elitism:

In this method firstly the best chromosomes or few best chromosomes are copied to new population and the rest is done in classical way[4].It has the advantage of very rapidly increasing the performance of ga because it prevents loosing the best found solutions.

$$\Phi i = (F_{max} - F_{min}) - F_i(X)$$

B. Crossover:

The aim of crossover operator is to search the parameter space in such a way that the information stored in present string is maximally preserved because these parent strings are instances of good strings selected during reproduction[5]. It is a recombination operator that can be performed in three steps:

1. The reproduction operator selects a random pair of two individual strings for mating,
2. A cross-site is selected at random along the string length and
3. The position values are swapped between two strings following the cross-site.

Crossover can be performed in six ways:

- Single-site crossover**:**

A cross-site is selected randomly along the length of mated length of the mated strings and bits next to the cross sites are exchanged. Choice of site must be done in better way to obtain better children by combining goo substances of parents. Sit is selected randomly.[8]

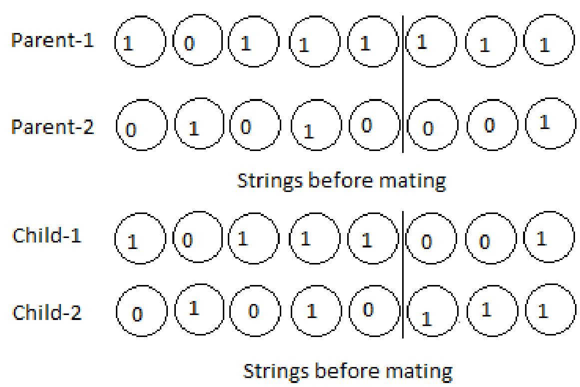

Fig6 Showing Single-site Cross-over

- Two-point crossover**:**

Here two random sites are chosen and the contents bracketed by these sites are exchanged between two mated parents. [7]If the cross-site 1 is three and the cross site 2 is six, strings between three and six will be exchanged as:

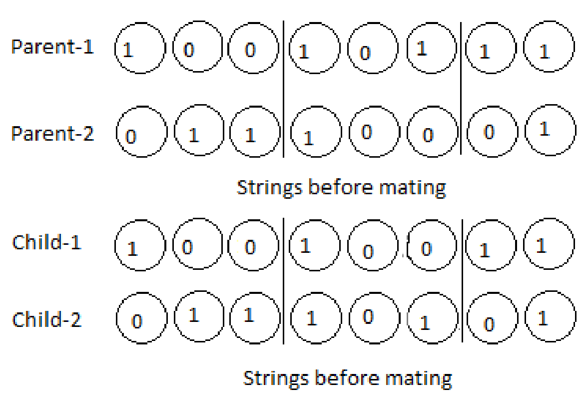

Fig7 showing Two-point Crossover

- Multi-point crossover:

Here we have two cases. In one case even number of cross-sites are considered and in other odd number of cross-sites. The string is treated as a ring with no beginning or end in even case and the cross sites are selected around the circle uniformly at random. Then the information between alternate pair of sites is interchanged as shown in figure:

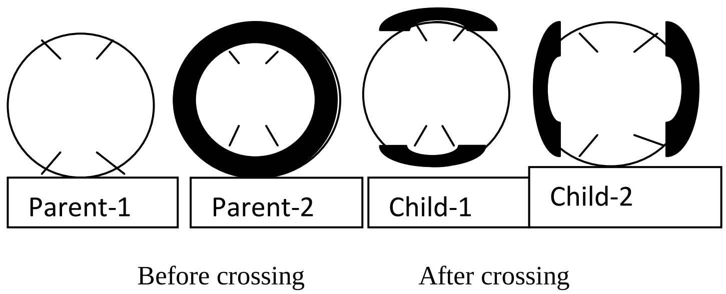

Fig8 showing Multi point Crossover with even number of cross-sites

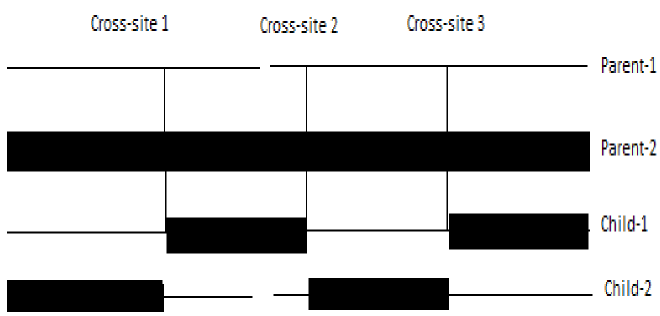

Fig9 showing Multi point Crossover with odd number of cross-sites

If the number of sites is odd, then we assume a different cross point at the beginning. The information between alternate pairs is exchanged as shown in figure.

- Uniform crossover:

In this gene from either parent is selected with a probability of 0.5 or then it is interchanged. When there is a 1 in the randomly generated crossover mask, the gene is copied from the first parent and when this is 0 gene is copied from second parent. The number of effective crossover points is not fixed but averages to L/2 where L is chromosome length.[12]

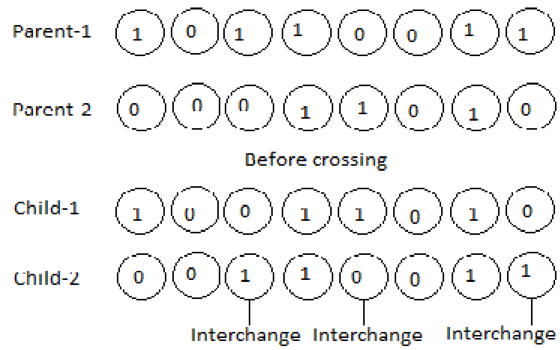

Fig10 showing Uniform Crossover

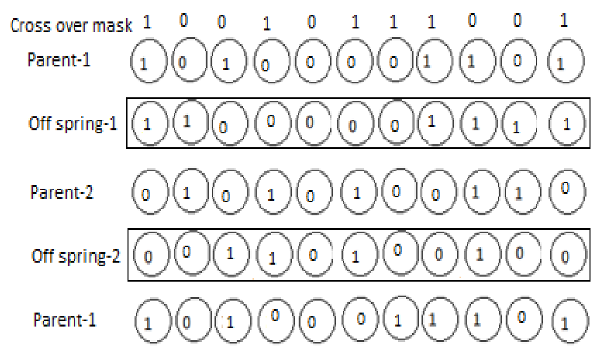

Fig11 showing Uniform Crossover using mask

- Matrix crossover:

Here two strings of length 4 are concatenated to form an individual. We select single dimensional cross-sites here and for two dimensional crossover we represent each individual as a two dimensional array of vector as shown below in example:[10]

String-1	1 0 1 1	1 0 0 1
String-2	0 1 0 1	1 1 1 0
	Substring-1	Substring-2

C. Mutation operator:

The aim of mutation operator is to restore lost genetic materials. It introduces new genetic structures in the population by randomly modifying some of its building blocks. It eliminates the problem of local minima of an algorithm. It is also used to maintain diversity in population. It causes movement in the local or global search space and is capable of restoring lost information to the population as shown below:[8]
E.g. : Population having two eight bit strings is shown:

0 1 1 0   1 0 1 1
0 1 1 1   1 1 0 0

To create a one in left most position for true optimum, then this job cannot be performed by crossover or reproduction, only mutation operator can perform this with some probability(Npm) as:

0 1 1 0   1 0 1 1
1 1 1 1   1 1 1 1

Pm is the mutation rate that can be defined as the probability of mutation to calculate number of bits to be muted.[16]

D. Regeneration:

The number of generation is typically around 50, this has been sufficient to arrive at a global optimum in most of the problems that use genetic programming. The depth of the tree structure also depends on the complexity of the problem and also the resources available, such as memory.[11] There are no rules to describe what parameters should be used and often it is simply best to try different options and see which best suits a particular problem. It can be considered something of a creative exercise.[9]

III. PROPOSED WORK

Population is generated using Rank Selection method. Effort has been applied for the development of Hybrid Genetic Algorithm. Hybridization has been suggested with the use of Fuzzy logic with the genetic approach.[14] Fuzzy has been applied here with the crossover function. Proposed algorithm is the Hybrid Genetics algorithm that provided us an optimized solution for the Travelling Salesman problem. According to this algorithm at first the population is defined in terms of nodes or the cities that placed at random position. Now the heuristic is applied on it to find the Seed Sequence. Now using this sequence a new population is generated. Each population is derived from the existing population. On this population the algorithm is performed. [13]

A. TSP verses Genetics:

- The proposed algorithm is given below:

TSP (City Matrix)
/*City Matrix is the actual distance matrix */
{
1. Perform the basic analysis over the City Matrix
to identify the maximum and minimum distance constraints
2. Generate the fuzzy rules based on the distance
Analysis
3. Perform the Matrix Quantization using fuzzy
vectors.
4. Define the initial fitness function.
5. Generate the N permutations of possible
Travelling path and represent it as the initial population set.
For i=1 to Max Iteration [Repeat steps 6 to 10]
6. Perform the selection on this training dataset.
7. Perform the fuzzy based crossover on selected
parents and generate the next level child.
8. Perform the mutation to neglect the values that
does not support fitness function.
9. Recombine the generated child with existing
population to generate new population set.
10. Identify the optimized Travelling salesman path
and present as result.[15]
}

This work contains certain assumptions and stages described below.

B. Assumptions:

- Cities are placed at random.
- Bi-directed path has been taken.
- No city can be visited more then once.

- Graph may not be a complete graph.[12]

C. Stages:

- This population has used Genetics for input.
- Fitness function is calculated. Min function is used for finding distance.
- Then rank selection method is applied for selecting the best individuals.
- Finally mutation function is applied on selected individuals.[6]

IV. CONCLUSION

In this paper, a genetic based solution is provided to find the solution of TSP problem in effective time. The paper has defined the genetic approach with the exploration of different stages. The main stress in this paper is given to the different approaches of selection and crossover process. The paper also included the proposed algorithmic model to obtain the effective solution of TSP.

REFERENCES

1. Whitley et al. (1990*), "Traveling Salesman and Sequence Scheduling: Quality Solutions using Genetic Edge Recombination"*, in Handbook of Genetic Algorithms, pp. 350-372.
2. Michalewicz et al. (1990), "*A Nonstandard Genetic Algorithm for the Nonlinear Transportation Problems*", ORSA Journal on Computing, Vol. 3, pp. 307-316.
3. Prasanna Jog, Jung Y. Suh et al. (1991)," *Parallel genetic algorithms applied to the Traveling Salesman Problem*", SIAM Journal of Optimization, Vol.1, pp. 515– 529.
4. Ulder et al. (1991),"*Genetic Local Search Algorithms for the Traveling Salesman Problem*", Lecture Notes in Computer Science, Vol. 496, Springer-Verlag, pp. 109-116.
5. H.Braun (1991),"*On Solving Travelling Salesman Problems by Genetic algorithms* ",Lecture Notes in Computer Science, Vol. 496, Springer-Verlag, pp. 129-133.
6. Abdollah Homaifar et al.(1992)," *Schema analysis of the Traveling Salesman Problem using Genetic Algorithms*", Complex Systems, 6(2), pp.183–217
7. Gerard Reinelt (1994)," *The Traveling Salesman: Computational Solutions for TSP Applications*", Lecture Notes in Computer Science, Vol. 840, Springer-Verlag.
8. D. Applegate, R. Bixby et al. (1998),"*On the solution of the traveling salesman problem*" *Doc*umenta Mathematica, Extra Volume ICM III, pp. 645–656.
9. Huai-Kuang Tsai, Jinn-Moon Yang et al. (2004), "*An Evolutionary Algorithm for Large Traveling Salesman Problems*" IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, Vol. 34, No. 4, pp.1718-1729.
10. Gerard Reinelt (1994)," *The Traveling Salesman: Computational Solutions for TSP Applications*", Lecture Notes in Computer Science, Vol. 840, Springer-Verlag.
11. Yuping Wang and Jinhua Qin (2007),"*A Memetic-Clustering-Based Evolution Strategy for Traveling Salesman Problems*" Lecture Notes in Computer Science, Vol. 4481, Springer-Verlag Berlin Heidelberg , pp. 260–266.
12. Rajan K and A. kumar (2009),"*Genetically Motivated Search Algorithm For Solving Travelling Salesman Problem*" ISSN: 0973-7464 Vol. XVI: No. 1 & 2 SB Academic Review, pp.19-31.
13. A.Chaudhuri and K. De, (2011), "*Fuzzy multi-objective linear programming for traveling salesman problem*", African Journal of Mathematics and Computer Science Research Vol. 4(2), pp. 64-70.
14. Fozia Hanif Khan, Nasiruddin Khan, Syed Inayatulllah, Shaikh Tajuddin Nizami and Muhammad Imtiaz(2011), "*Hybrid approach for solving TSP by using DPX Cross-over operator*", in Pelagia Research Library Advances in Applied Science Research, 2011, 2 (1): 28-32
15. E.J. Teoh et al. (2008)," *An asynchronous recurrent linear threshold network approach to solving the Traveling Salesman*" Neurocomputing ,Volume 71, Issue 7 – 9, pp. 1359 – 1372.
16. Devasenatipathi N.Mudaliar, Dr. Nilesh K.Modi , " *Unraveling Travelling Salesperson Problem By Genetic Algorithm Using Mcrossover -Operator*" 2013 International conference on Signal Processing, Image Processing and Pattern Recognition [ICSIPR].