Genetic Approach to Travelling Salesman Problem

Under the supervision of Mr. Radha Krishna Rambola(Assistant Professor)

Submitted By:

Binit Kumar

APIIT SD INDIA, Panipat, Haryana

ABSTRACT:

This paper includes an efficient method to solve travelling salesman problem by using genetic approach of solving. The domain problem in this paper is TSP. TSP has always known as a standard and historical example of NP-complete problems. There had been much different approach to solve this with different success rate. This paper provides the solution which includes genetic algorithm implementation in order to give a maximal approximation of the problem with the reduction of cost. In genetic algorithm crossover is the major operator that do the main solving and optimization. There were various attempts to find the most suitable mating algorithm in GA. This paper includes and defines the strategy to find the optimisation of problem using the efficient crossover operator that research paper on travelling salesman problem using genetic algorithms that results in high quality solution to the TSP. The strategies proposed here intend to compare various crossover operators and find the most efficient to the TSP.

Keywords: *Genetic algorithm, Travelling Salesman Problem, Crossover operator, Mutation*

1. INTRODUCTION:

The Travelling Salesman problem (TSP):

The Travelling Salesman Problem (TSP) is a classic combinatorial optimization problem, which is simple to state but very difficult to solve. This problem is known to be NP-hard, so it cannot be solvable in polynomial time. The problem is to find the shortest possible tour through a set of n vertices so that each vertices is visited exactly once with common starting and ending point. Many exact and heuristic algorithms like simulated annealing have been developed in the field of operations research (OR) to solve this problem.

German Salesman BF Voigt`s book on how to be a successful travelling salesman is one of the first discussion that gained fame. He mentions suggestion to cover as many locations as possible without visiting any location twice. This is the most important aspect of the scheduling of a tour which gets popular as TSP. In mathematics world the origin of TSP is unknown but it comes around 1931.

TSP can be classified on the basis of cost matrix into two groups – symmetric and asymmetric. The TSP is symmetric if $C_{ij} = C_{ji}$, $\forall$ i, j and asymmetric otherwise. For an$(n-1)$ city asymmetric TSP, there are $(n-1)!$ possible solutions, one or more of which gives the minimum cost. For an n-city symmetric TSP, there are$\frac{(n-1)!}{2}$ possible solutions along with their reverse cyclic permutations having the same total cost. The major problem is that in either case the number of trial grows in order but the number of solutions grows exponentially (e^x). It results in impracticable exhaustive search even in the case of supercomputers.

Genetic algorithm (GA):

Genetic algorithm (GA) as a computational intelligence method is a search technique used in computer science to find approximate solutions to combinatorial optimization problems. Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic.

The basic concept of GAs is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin of survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem.

The genetic algorithm process consists of the following:

Encoding: A suitable encoding which is unique is found for the solution may be binary or continuous in some form of string. Data structures can be used in programming languages.

Evaluation: The GA starts with a group of chromosome known as population. The population has N_{pop} chromosomes. The fitness of each individual in the population is computed using cost function.

Crossover: The fitness is used to create new offspring from the parents selected in pairing process. The new offspring are obviously of better fitness and copied into the new generation.

Mutation: Random mutation alters a certain percentage of bits in the list of chromosomes. Mutation is the second way a GA explores a cost surface.

Decoding: Once this is done, a new generation has been formed and the process is repeated until some stopping criteria have been reached. Convergence Check is done to find an acceptable solution

2. GENETIC APPROACH TO TSP

To solve the complex NP-hard problems GA can be used to find the solution in very less time. The solution cannot be guaranteed as the best suitable solution, but it is near to optimised solution.

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, pick 2 of the better (shorter) tours parents in the population and combine them to make 2 new child tours. Hopefully, these children tour will be. A small percentage of the time, the child tours is mutated. This is done to prevent all tours in the population from looking identical. The new child tours are inserted into the population replacing two of the longer tours. The size of the population remains the same. New children tours are repeatedly created until the desired goal is reached.

After comparing these factors in each solution the best one will be selected and hence will give the new shortest path in each iteration. The task of comparisons and then representing the solution in every iteration become complex with the increment in population size

3. RELATED WORK

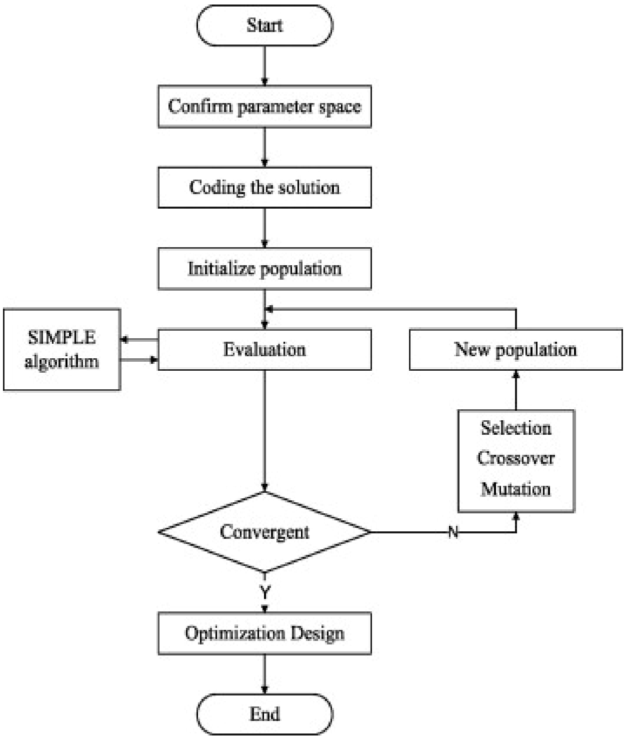
In the work proposed by Omar M.Sallabi An Improved Genetic Algorithm to Solve the Traveling Salesman Problem[1] profound The Genetic Algorithm (GA) is one of the most important methods used to solve many combinatorial optimization problems. Therefore, many researchers have tried to improve the GA by using different methods and operations in order to find the optimal solution within reasonable time. This paper proposes an improved GA (IGA), where the new crossover operation, population reformulates operation; multi mutation operation, partial local optimal mutation operation, and rearrangement operation are used to solve the Traveling Salesman Problem. Crossover is the most important operation of a GA because in this operation, characteristics are exchanged between the individuals of the population.

In the work proposed by Zakir H. Ahmed Genetic Algorithm for the TSP using Sequential Constructive Crossover Operator[2] develops a new crossover operator, Sequential Constructive crossover for a genetic algorithm that generates high quality solutions to the Traveling Salesman Problem (TSP) . Genetic algorithms (GAs) are based essentially on mimicking the survival of the fittest among the Species generated by random changes in the gene-structure of the chromosomes in the evolutionary biology. To apply GA for any optimization problem, one has to think a way for encoding solutions as feasible chromosomes so that the crossovers of feasible chromosomes result in feasible chromosomes. For the TSP, solution is typically represented by chromosome of length as the number of nodes in the problem

4. METHODOLGY:

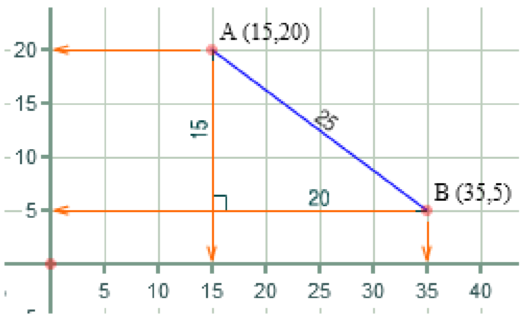
A simple GA works by randomly generating an initial population of strings, which is referred as gene pool and then applying (possibly three) operators to create new, and hopefully, better populations as successive generations. The first operator is reproduction where strings are copied to the next generation with some probability based on their objective function value. The second operator is crossover where randomly selected pairs of strings are mated, creating new strings. The third operator, mutation, is the occasional random alteration of the value at a string position. The crossover operator together with reproduction is the most powerful process in the GA search. Mutation diversifies the search space and protects from loss of genetic material that can be caused by reproduction and crossover. So, the probability of applying mutation is set very low, whereas the probability of crossover is set very high.

Steps of Algorithms:



Step 1: Cost Function: The cost function id determined by the public sub procedures in Cities class. It is basically derived from the Pythagoras Theorem of calculation that is $Distance = \sqrt{(x - x1)^2 + (y - y1)^2}$

Where x, y is next city's location and x1,y1 is current city location.



The City distances are calculated for all the cities and further the distance are added to the array list for processing.

Step 2: Closet city with minimum distance logic: When creating the initial population of tours, this is a greater chance that a nearby city will be chosen for a link. This is the number of cities that will be considered close

```
1.START
2. FOR EACH i
2.1 FOR EACH cityNum
2.1.1 IF distance of cityNim is smaller then shortest distance THEN
2.1.2 shortest distance EQUALS distance of cityNum
2.1.3 shortest city is cityNum
2.1.4 END IF
2.2 END FOR
2.3 ADD shortest city of closecities array
2.4 END FOR
```

The final array include the array of the shortest distance to the next city of the all the distances of that city to all other cities. The final values stored in the process are the cities that have the shortest distance

Step 3: Map the chromosome: The chromosome is set of ordered indexes of cities, through which the traveller goes. The TSP is coded into data structure, which can be handed like a chromosome. The array index is our route order, the contents of each array elements is a city number. Hence, if developer wrote the following pseudo code:

A[ ] = { 5 ,64, 23 ,8 , 32 …….. }

It would mean that the visit of city 5 is the starting point followed by city 64, 23, 8, 32 and so on. This is how 'chromosomes' or trial solutions is encoded in genetic algorithm. In the code, tour class represents the order of cities linked together and object of the tour class is used to find fitness and other processing in GA.

Step 4: Fitness Function: The fitness is calculated for each chromosome that further mates to create new offspring. Fitness is the criteria of ranking the tours. TSP determines fitness as the total distance of a Tour, Root Mean Square (RMS) value is used to determine fitness from the cost function the higher the distance, lower is the fitness.

$$Fitness = \frac{1}{Total\ Distance}$$

Step 5: Genetic algorithm engine for population control: To begin a GA, we define an initial population of N_{pop} chromosomes. A matrix represents the population with each row in the matrix being a 1 * N_{var} array (chromosome) of continuous values.

$$\boldsymbol{pop = rand(N_{pop}, N_{var})}$$

If initial look at this problem with N =13 cities with the fixed starting and ending points, there are a total of $\boldsymbol{\frac{13!}{2}}$ = **3.1135 *$\boldsymbol{10^9}$** possible combinations can be generated. In the Implementation, developer has provided indefinite number of cities to select for and this results in a very large number of possible and unexpected tours.

```
Step1. START
Step2. CREATE Class population
Step 3. Inherit arrayList of tour
Step 4. SAVE private bestTour as NULL
Step 5  GET/SET bestTour
Step 6. Generate Random Population
Step 7. UPDATE bestTour
Step 8.  CREATE tour of length of city list
Step 9.  Randomly select first city
Step 10.  FOR EACH city
Step 10.1.  DO
Step 10.2.  IF  new next city , ADD
Step 10.3  ELSE ADD random next city
Step 10.4  .END WHILE when next city is last city
```

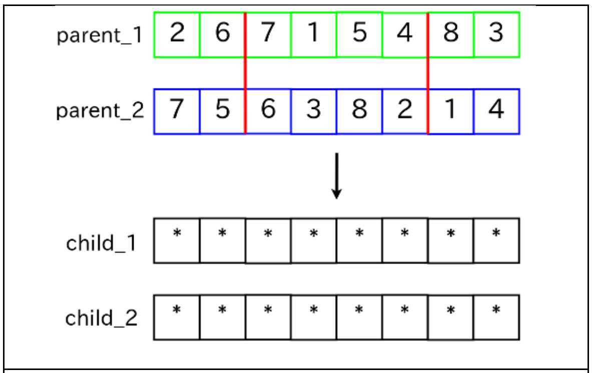
Step 10.5 END FOR
Step 11 Connect last two cities using
Step 12 Determine tour fitness
Step 13. ADD tour in the Population
Step 14. IF tour fitness is smaller than best tour fitness
Step 15. REPLACE best tour with tour
Step 16 STOP

Step 6: Crossover: Crossover comes after the generation of initial population the GA flowchart. Crossover is done to the parents to create new offspring of better and hybrid cost.

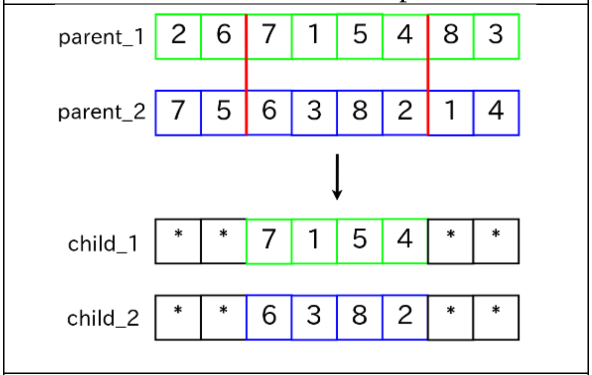
Objective:

- Perform the crossover operation on two parent tours to create new child tour
- Total of two children will be created so twice execution of the function
- Count links of the cities using iteration
- Take all the common links from both parents and put them in child, this will inherit parent's traits into child tours.
- Randomly assign remaining links as parent's link cause multiple disconnected loops in the tour

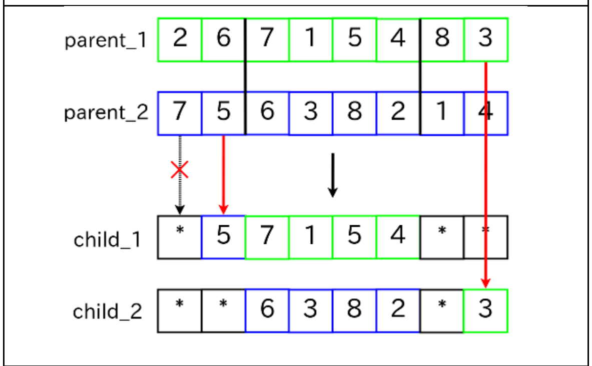
The crossover technique used here is much similar as PMX (partially mapped Crossover (**Goldberg, 1985**). A better crossover should introduce new edges to the children and common edges must be inherited. Common edges preserve parent's traits and new edges provide diversity and escape from local optima



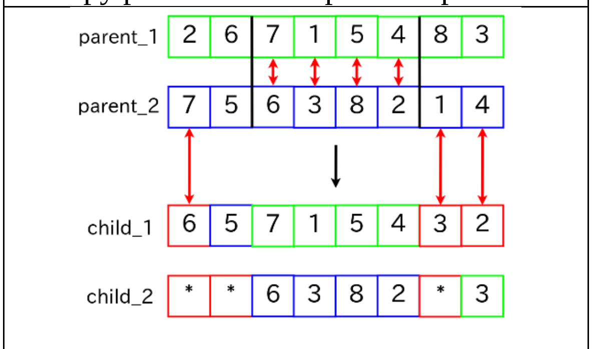
Decide two crossover points



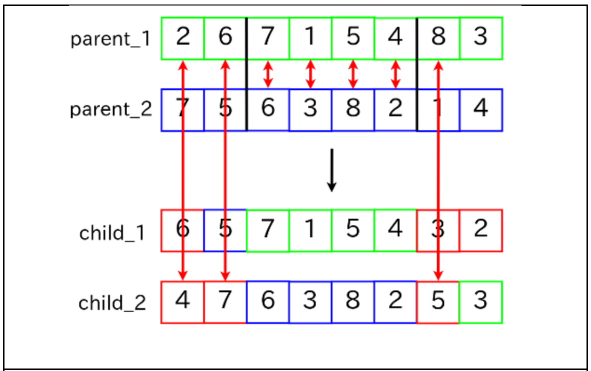
Copy path between two crossover point



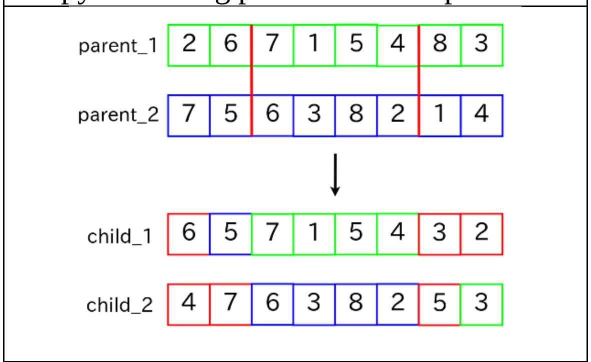
Copy path of another parent if possible



Copy remaining path with correspondence

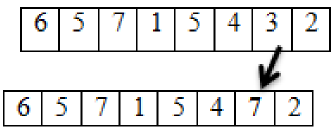


Copy remaining path with correspondance

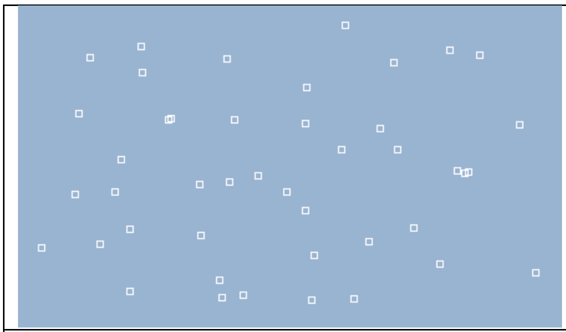


Result of PMX

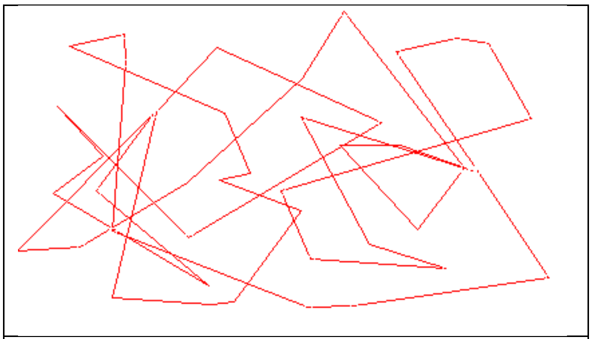
Step 7: Mutate: Random mutations are executed side by side of mating algorithm (crossover) for moving the population from local extrema. It could happen that the computation is in such a state locked in non-optimal position and needs external (random) impulse to break the disability apart an start again moving ahead. On the other hand, if there is too much of random process into computation, the result will never be the most optimal.



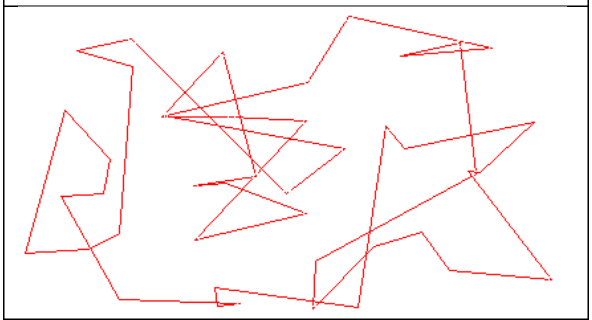
Output



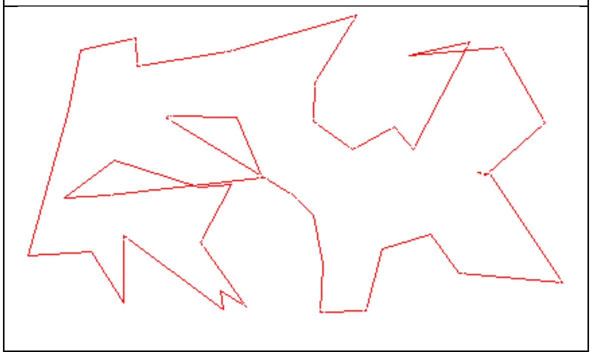
TSP : 42 cities, Distance : unknown



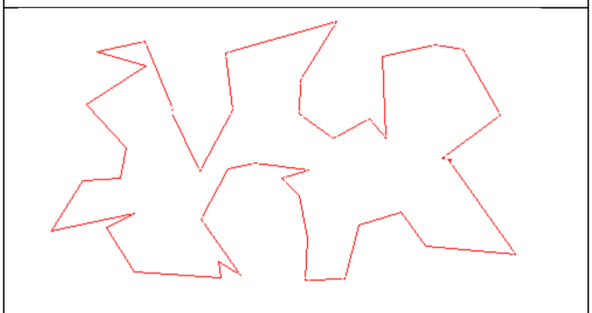
Tour 1: distance : 5582



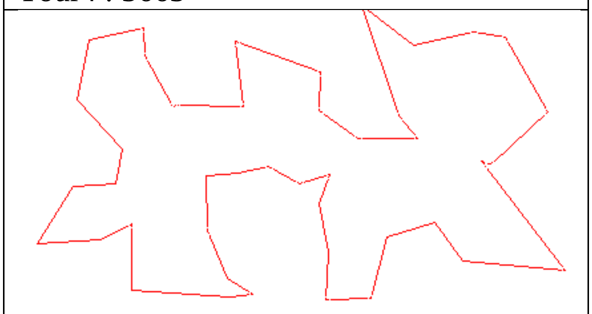
Tour2 : distance :4354



Tour3: distance 3823



Tour4 : 3663



Tour5: 3390

5. RESULT

The results above are the five tours that come on the basis of our calculation through the application based on the same defined approach as a Final Year Project by Researcher. The results show that how we can reach the approximately optimal path for TSP using genetic algorithm in very less time. Above results of shortest route comes after 30000 generation with initial mutation rate of 3% , no of close cities 5% and random seed 5 . Cities are randomly assigned at different random co-ordinates. With the mutation the mating happens at random location rather than sequentially search for each gene in a chromosome. Hence after comparing the fitness of parents and offspring's , the new offspring comes in the population with better fitness of Tour.

Algorithm	**Time**	**Complexity**	**Advantage**	**Disadvantage**
Genetic Algorithm	**Exponential time**	O(kmn)	**Best Solution using fitness criteria**	**Approximation of Solution is reached but no optimal solution**
Greedy Approach	**5 seconds for 15 cities**	O(log n)	**Fastest**	**No best choices are considered , so no accuracy acheived**
Dynamic Programming	**9 seconds for 15 cities**	$On^2 2^n$	**Global Optimal Solution**	**Expensive for memory and time**

6. CONCLUSION AND FUTURE WORK

Genetic algorithm is one of that typical solution that follows that nature rule of Survival of the Fittest and seems perfect, however it depends very much on the way the problem is encoded and which crossover and mutation methods are used. The paper has proposed new genetic algorithm approach which solves problem continuously rather than by binary approach done through using each bits. Mutation is also solved in percentage and a new approach of close cities is introduced. The paper set here highest probability of crossover to show the exact nature of crossover operator.

The paper also presented a comparative study among diferent approaches of solving TSP. A good local search technique is required is the fun part in the modern coding and mathematics. Most of the problem like circuit design, pipelining etc can be approached using this paper and solving style through GA.

7. REFERENCES

1. Omar M.Sallabi, " *An Improved Genetic Algorithm to Solve the Travelling SalesmanProblem"* World Academy of Science,Engineering and Technology 52 2009

2. Zakir H. Ahmad, " *Genetic Algorithm for the TSP using Sequenctial Constructive Crossover Operator"* International Journal

of Biometrics and Bioinformatics(IJBB) Volume (3): Issue (6)

3. Bajpai, 2008, *Genetic Algorithm – an Approach to Solve Global Optimization Problems.* Indian Journal of Computer Science and Engineering, 1-3.

4. Haupt, 2004, Practical Genetic Algorithms, 2nd Edition, The Binary Genetic Algorithm (p. 28), John Wiley & Sons, Inc, New Jersey

5. Ray, Bandyopadhyay, Pal,2004, New Operators of Genetic Algorithm for Travelling Salesman Problem, Indian Statistical Institute, Kolkata

6. Eric matel , "Solving the travelling salesman problem using a genetic algorithm".

7. Faulkner, G. Talhami, H. ,Kalgoorlie Coll.,WA, "Using biological genetic algorithm tosolve the travelling salesman problem with applications in medical engineering",Evolutionary Computation, 1995., IEEEInternational Conference in 1995.

8. Geetha, R.R.; Bouvanasilan,N.; Seenuvasan, V.,"A perspective view on Travelling Salesman Problem using genetic algorithm" Nature & Biologically Inspired Computing, 2009. NaBIC 2009, World Congress.