



**EFFICIENT TECHNIQUE TO SOLVE TRAVELLING SALESMAN
PROBLEM USING GENETIC ALGORITHM**

Prof. Sharadindu Roy, Uttam Kumar Panja, Nikhil Kumar Sardar

Sonarpur Mahavidyalaya, University of Calcutta

ABSTRACT

In this paper, an efficient technique is proposed to solve the **travelling salesman problem** (TSP) using genetic algorithm. We can apply easily genetic algorithm operator to this problem and get the solution easily. Complexity is both in time and space, provided size of the problem an as integer. The solution of the traveling salesman problem is global optimum. There are number of cities and distances are given between them (cities). A Traveling salesman has to visit all of them. The salesman will start from city and after traversing the remaining cities (exactly once) he will reach to his starting position. Main objective of TSP is to find traveling sequence of cities to minimize the traveling distance so that the salesman can traverse the cities exactly one time. Initially we select parent1 & parent2 by Roulette wheel selection philosophy. We have applied one point crossover operator on the parent and produce the new child. Again we apply the mutation operator on offspring and created new child. But the no. of bits (cities) in a chromosome will be inverted by the mutation operator, that is depended on mutation probability (p_m). So, one generation contain 6 individual (chromosome). We have to count fitness (minimum cost) of the individuals in each generation. We have to select two individuals with best (min fitness) fitness for the next generation. Here we see crossover between two good solution may not always yield a better or as good a solution. But since the parents are good, so there is a probability that child will be good. Every time we have to do, identity the good solution in the population and make multiple copies of the good solution.

Keyword: Genetic algorithm, Crossover, Mutation, Travelling salesman problem, NP-complete, Cost matrix, Fitness.

INTRODUCTION

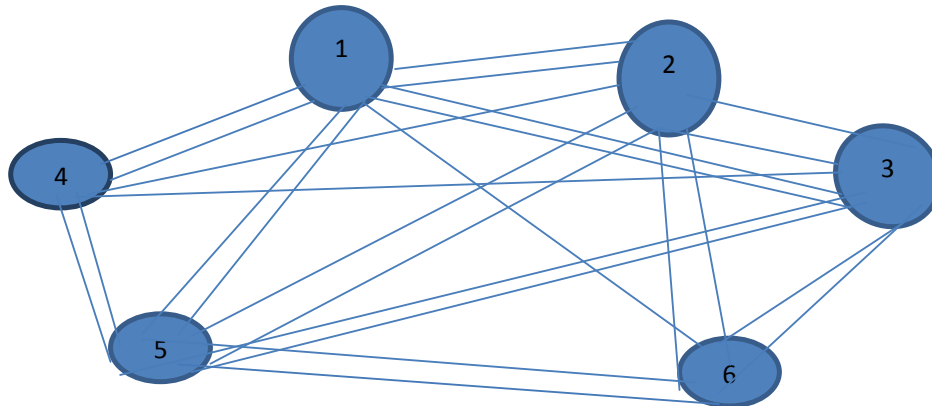
Travelling salesman problem (TSP) is one of the old problems in computer science and operations Research. This problem is: A graph with ‘n’ nodes (or cities), with ‘node 1’ as a

'headquarters' and travel cost (or distances, or travel time etc.,) matrix $C=[c_{ij}]$ of order n associated with ordered node pairs (i, j) is given. The problem is to find a minimum cost Hamiltonian cycle. The search space contains $N!$ Permutations and since TSP is NP-complete and the corresponding optimization problems are therefore NP-hard. The problem with this representation is obvious. Starting with a population of valid chromosomes, ordinary crossover and mutation operators cause problems. In this algorithm we use the one point crossover operator but mutations are not performed at single points. Here, simple bit-string crossover and

Ideas related to the TSP have been around for a long time: In 1736, Leonard Euler studied the problem of finding a round trip through seven bridges in Königsberg. In 1832, a handbook was published for German travelling salesmen, which included examples of tours. In the 1850s, Sir William Rowan Hamilton studied Hamiltonian circuits in graphs. He also marketed his 'Icosian Game', based on finding tours in a graph with 20 vertices and 30 edges.

Solution methodology

Let us consider the following graph containing six nodes(cities). Two edges between each pair of vertices denotes different distance. The distance between each pair of cities are not shown.



For the TSP, solution is typically represented by chromosome of length as the number of nodes in the problem. Each gene of a chromosome takes a label of node such that no node can appear twice in the same chromosome. There are mainly two representation methods for representing tour of the TSP - adjacency representation and path representation. We consider the path representation for a tour, which simply lists the label of nodes. For example, let $\{1,2,3,4,5,6\}$ be the label of nodes in 5 node instances, then a tour $\{1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 1\}$ may be represented as $(1,3,4,2,5,6)$. At first the parent1 & parent2 will be selected via roulette wheel selection concept. Here we apply crossover operator that will be - part of the first parent is copied and the rest is taken in the same order as in the second parent.

For example:

The 6 cities- 1) Kolkata 2) Chennai, 3) Bhubeneswar 4) Bengaluru, 5) Hydra bad, 6) Tirupati.

We take the any sequence of the 6 cities, that can be-

- 1 3 6 2 4 5
- 3 2 1 5 6 4
- 2 5 3 4 6 1
- 1 5 3 2 4 6

If, fitness of the chromosomes are respectively 45, 43, 11, 78 then roulette wheel select 3 and 2 no. chromosome. Suppose the chromosome “1 3 6 2 4 5” , For this chromosome the fitness will be calculated as a following process-----

Fitness=1 to 3 distance + 3 to 6 + 6 to 2 + 2 to 4 + 4 to 5 + 5 to 1 distance.

Algorithm:

- Step1: start with initial population
- Step2 select parent1 & parent2 (via roulette wheel selection)
- Step3 : Define crossover type & mutation probability
- Step4: Apply crossover operator on parents (created offspring)
- step5: Apply mutation operator on parents (created offspring)
- step6: If optimum result is found then go to step 9 else goto step7
- step7: Chose two best fitness chromosomes for next generation and go to step4
- step8: if fitness of created offspring are equal for some generation then change crossover boundary and mutation probability and go to step 4
- Step9: stop

Experimental result:

There are 8 cities. The travelling cost between cities shows is in table that is called cost matrix. Cost matrix =

	1	2	3	4	5	6	7	8
1	0	50	75	25	40	30	60	80
2	40	0	25	50	30	40	80	60
3	75	40	0	30	80	50	25	40
4	50	30	60	0	60	25	40	25
5	30	25	40	40	0	75	75	40
6	25	80	30	80	25	0	60	30
7	80	60	50	60	75	40	0	75
8	60	75	80	25	40	60	30	0

The genetic algorithm has been used the minimized the travelling cost between many cities. The coding has been done using MATLAB r2012a (7.3 versions). The above example contains 8 cities, and the proposed algorithm is applied by the MATLAB r2012a on above cities and run the

programme and takes the result after 10 generation. Three graph are shown in the following after running the program in matlab. Fig1. Shows plotting for chromosome vs. MINIMUM_fitness in 100 generation. Fig2. Shows plotting for chromosome vs. maximum_fitness in 100 generation and fig3 Shows plotting for chromosome vs. maximum_fitness in 100 generation.

The genetic algorithm has been used the minimized the travelling cost between many cities. The coding has been done using MATLAB r2012a (7.3 versions). The above example contains 8 cities, and the proposed algorithm is applied by the MATLAB r2012a on above cities and run the programme and takes the result after 10 generation. The result are shown in tabular form

No. of generation	Minimum fitness(distance)	Maximum fitness(distance)	Average fitness(distance)
10	350	495	387
20	310	380	340
30	270	425	323
40	270	380	316
50	270	415	318
60	270	350	307
70	270	325	292
80	270	435	327
90	270	445	326
100	270	435	327

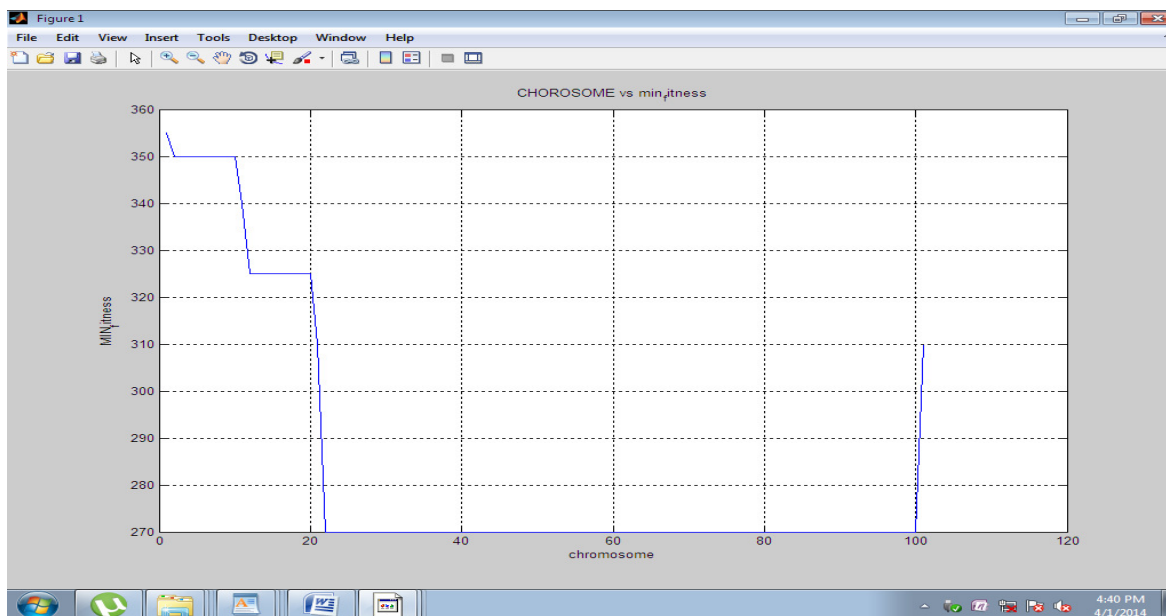


Fig1: plot for chromosome vs. MINIMUM_fitness in 100 generation

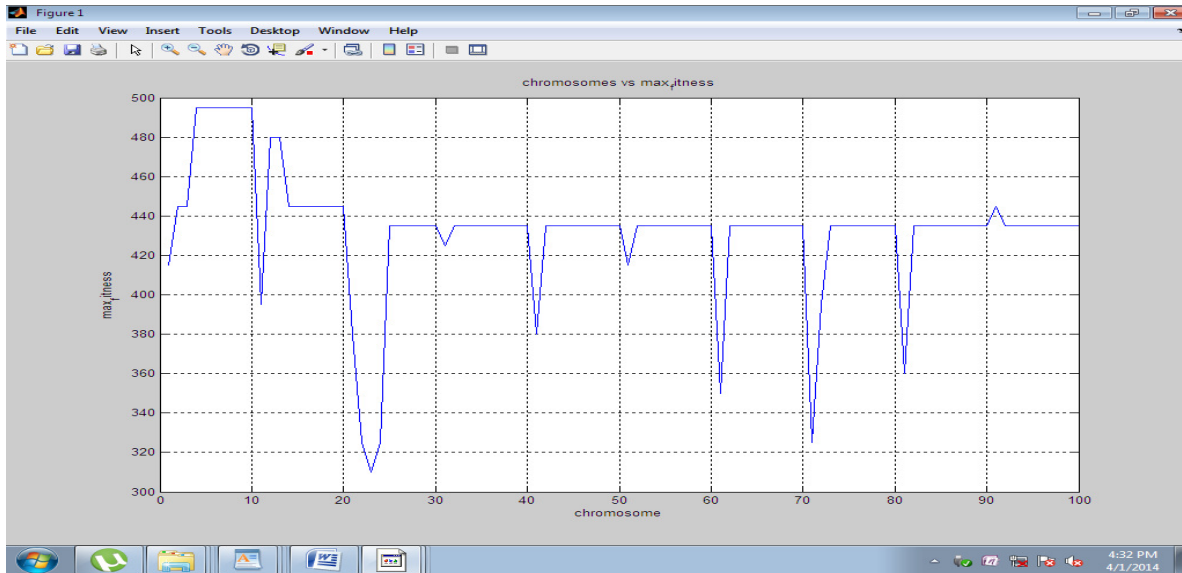


Fig2: plot for chromosome vs. maximum_fitness in 100 generation

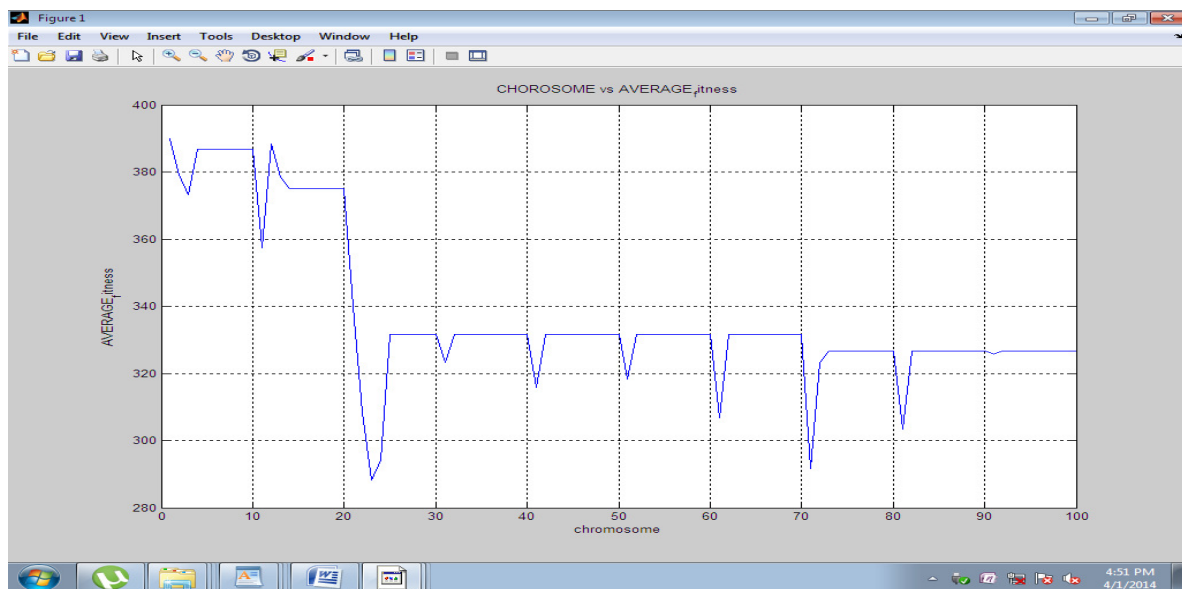


Fig3: plot for chromosome vs. AVERAGE_fitness in 100 generation

As seen from the results in Fig.1,fig.2 and fig3 when increasing the no.of iteration 1:10 we get desired minimum fitness in first 10th generation that is 350 , maximum fitness with value 495 and avarage fitness is 387. When the no. of iteration are 11 to 20, change the crossover boundary and mutation probability (Pm=0.5) then the minimum fitness is 270. After a certain generation, when we change the crossover boundary and mutation probability (Pm=0.25) in generation 51 to 60, we get desired minimum fitness better then the previous generation. The fig.5 reflex the iteration no. 51 to 60.So, we see that, fitness is not dependent on crossover boundary and mutation probability but there are probability. when iteration number 81 to 90 , then we change the crossover boundary and mutation probability (Pm=0.75), we get the minimum cost between 1 to 100 generation, that 270. Minimum fitness,avaraage fitness and max fitness are shown respectively in Fig.1 ,fig3,fig2 whose iteration number between 1to 100.

CONCLUSION

Charles Darwin's principle of natural selection is followed here. We have followed to select the best and discard the rest chromosomes. Main philosophy of genetic algorithm is followed to Holland. We apply one-point crossover operator for a genetic algorithm for the travelling salesman problem. In this method, when we get the minimum cost in a generation then we will see that minimum cost is also in next generation or reduce the minimum cost of previous generation. So, minimum cost is easily found in many generations.

REFERENCE

- [1] D.E. Goldberg and R. Lingle. "Alleles, Loci and the travelling salesman problem". In J.J. Grefenstette (ed) proceedings of the 1st International Conference on genetic algorithms and their applications. Lawrence Erlbaum, Hilldale, NJ, 1985.
- [2] Davis. "job-shop Scheduling with Genetic algorithms". Proceeding of International Conference on genetic algorithms and their applications, pp. 136-140, 1985.
- [3] I.M. Oliver, D. J. Smith and J.R.C. Holland. "A Study of Permutation Crossover Operators on the Travelling Salesman Problem". In J.J. Grefenstette (ed.). Genetic Algorithms and Their Applications: Proceedings of the 2nd International Conference on Genetic Algorithms. Lawrence Erlbaum Associates, Hilldale, NJ, 1987.
- [4] D.E. Goldberg. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, New York, 1989.
- [5] R.G. Bland and D.F. Shallcross. "Large Travelling Salesman Problems arising from Experiments in X-ray Crystallography: A Preliminary Report on Computation". Operations Research Letters 8, pp. 125-128, 1989.
- [6] D. Whitley, T. Starkweather and D. Shaner. "The Traveling Salesman and Sequence Scheduling: Quality Solutions using Genetic Edge Recombination". In L. Davis (Ed.) Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, pp. 350-372, 1991.
- [7] K. Deb. "Optimization For Engineering Design: Algorithms And Examples". Prentice Hall Of India Pvt. Ltd., New Delhi, India, 1995.
- [8] N.J. Radcliffe and P.D. Surry. "Formae and variance of fitness". In D. Whitley and M. Vose (Eds.) Foundations of Genetic Algorithms 3. Morgan Kaufmann, San Mateo, CA, pp. 51-72, 1995.
- [9] P. Poon and J. Carter. "Genetic algorithm crossover operations for ordering applications". Computers and Operations Research 22, pp. 135-47, 1995.
- [10] C.H. Papadimitriou and K. Steglitz. "Combinatorial Optimization: Algorithms and Complexity". Prentice Hall of India Private Limited, India, 1997
- [11] Z.H. Ahmed and S.N.N Pandit. "The travelling man salesman problem with precedence constraints". Opsearch 38, pp. 299-318, 2001.
- [12] C. Moon, J. Kim, G. Choi and Y. Seo. "An efficient genetic algorithm for the traveling salesman problem with precedence constraints". European Journal of Operational Research 140, pp. 606- 617, 2002.
- [13] I. Choi, S. Kim and H. Kim. "A genetic algorithm with a mixed region search for the asymmetric traveling salesman problem". Computers & Operations Research 30, pp. 773 - 786, 2003.
- [14] Gulwatanpreet Singh, Surbhi Gupta and Baldeep Singh, "Aco Based Solution for TSP Model for Evaluation of Software Test Suite", International Journal of Computer Engineering & Technology (IJCET), Volume 3, Issue 3, 2012, pp. 75 - 82, ISSN Print: 0976 - 6367, ISSN Online: 0976 - 6375.
- [15] Kulbhushan Verma, Manpreet Kaur and Palvee, "Comparative Analysis of Various Types of Genetic Algorithms to Resolve TSP", International Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 4, Issue 5, 2013, pp. 111 - 116, ISSN Print: 0976- 6464, ISSN Online: 0976 -6472.