Solving Travelling Salesman Problem Using Greedy Genetic Algorithm GGA

Vinod Jain[#1], Jay Shankar Prasad[*2]

[# 1]Ph.D. Scholar ,[*2] Professor

Department of Computer Science and Engineering,

MVN University, Palwal, Haryana, India

[#1] jainvinod81@gmail.com, [*2] jayshankar.prasad@mvn.edu.in

***Abstract*—** Travelling Salesman Problem represents a class of problems in computer science. This problem has many application areas in science and engineering. Genetic Algorithm is used to solve these problems and the performance of genetic algorithm depends on its operators. In this paper new greedy genetic algorithm has been proposed to solve TSP. The proposed greedy genetic algorithm is applied and tested on some standard TSP problems; the obtained results are compared with existing methods and found better in terms of path length. The proposed greedy genetic algorithm search deeper in the search space and find better solutions as compared to existing algorithms. The proposed algorithm finds solutions which are approximately 5% better than the existing algorithm.

Keyword- Travelling Salesman Problem, Genetic Algorithms, Greedy Approach

I. INTRODUCTION

Travelling Salesman Problem (TSP) is a well-known problem in computer science. It has many application areas in science and engineering. In TSP a hypothetical salesman has to visit a set of cities. Salesman start the journey from a city and visit each and every city exactly once. As the number of cities increases the number of possible paths also increases and the complexity of algorithm becomes n! if there are n cities. Evolutionary techniques such as Genetic Algorithm, PSO are very popular methods for solving NP-Complete and NP-Hard problems. In literature work has been done in solving Travelling Salesman Problem using Genetic Algorithm. Mei Mi et.el. [1] Proposed Liuhai cross over which includes the best individual preservation policy. Author generates the initial population using greedy approach. To generate the initial population, any city is selected randomly and then the remaining cities are selected according to the nearest neighbor approach. Author also selects the excellent chromosome to participate in the cross over. Shakeel Arshad, and Shengxiang Yang [2] try to solve TSP. Author divided the procedure into two phases. In phase 1 author uses SBGA technique. SBGA generate the initial population first and then uses embedded sequence based ordered cross over (e-SBOX) to generate new children. E-SBOX further uses SBLS to perform cross over. Phase 2 perform the modified inner over cross over algorithm. This phase perform restricted inner over with partial random initialization to generate new children with better fitness. Oliviu Matei and Petrica Pop [3] proposed a solution using GA for generalized travelling salesman problem. The proposed GA uses steady state approach. The chromosomes enter into the population as soon as they are produced and inferior chromosomes are removed from the population. Equal number of chromosome leave the population and equal number of chromosome enter into the population and the count or size of the population remains the same. The algorithm will make cluster of chromosomes. Ren Shuai, Wang Jing, Xuejun Zhang [4] proposed a genetic algorithm based solution for travelling salesman problem. The author perform the genetic operations in a sequence, It first generate the initial population, then initialize the genetic parameters, then perform Parthenon-genetic operators such as Gene swap, gene reverse, gene insert. Then author apply Chaos search, greedy local search and then calculate the fitness of all the chromosomes and then perform these operations in many iterations.

Abdoun Otman and Abouchabaka Jaafar [12] published a survey on genetic cross over operators. The paper compares the performance of various cross over operators of GA. From this discussion it is clear that the performance of Genetic Algorithm can be improved by improving cross over operator[12,13]. Hao Jia [11] proposed a hybrid optimization algorithm and test it for solving TSP. Author proposed a hybrid of Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) HPSACO for solving TSP. After implementation author concluded that the HPSACO algorithm performs better than the PSO and ACO. Gohar Vahdati, Sima Yaghoubian Ghouchani and Mahdi Yaghoobi [10] propose a hybrid search algorithm with Hopfield Neural Network HNN and Genetic Algorithm GA for Solving Traveling Salesman Problem. Author implements the hybrid algorithm and compare the result with some existing algorithms. Results show that the hybrid algorithm performs better than the existing algorithms.

II. PROPOSED WORK

In this paper a greedy genetic algorithm has been proposed. Algorithm -1 is a standard genetic algorithm to solve any problem. Steps of standard Genetic Algorithm are shown in Algorithm-1.

Algorithm-1: Standard Genetic Algorithm
This algorithm perform genetic algorithm on a given problem.

1:Encode given problem in genetic form.
2:Initialize GA parameters.
3:Generate initial population.
4:Select parents chromosomes from initial population to perform cross over.
5:Perform cross over to generate children.
6: Add new children in population.
7: Discard worst chromosomes from population.
8: Perform mutation.
9: Generate population for next generation.
10: If stopping criteria reached then STOP otherwise repeat steps4 to step9.

In this paper greedy approach has been used in genetic algorithm. In this work the proposed genetic algorithm is differ from standard genetic algorithm in three ways which are as follows:

- Greedy Initial Population
- Greedy Cross Over Operation
- Modified Mutation

Three changes made to classical genetic algorithm are Greedy Initial Population, Greedy Cross Over Operation and Modified Mutation. The following section explain the changes made in the classical genetic algorithm.

Greedy Initial Population – In this modified genetic algorithm the initial population is a hybrid of chromosomes generated using random approach and chromosomes generated using greedy approach. In first step chromosomes equal to the size of initial population has been generated using random population generation approach. Then a set of chromosomes are generated using nearest neighbor greedy approach and added in the initial population. Then the fitness of the complete initial population is calculated and best chromosomes equal to size of initial population has been taken. So in this way the greedy initial population has been generated. Algorithm to generate greedy initial population is shown in Algorithm-2.

Algorithm – 2 : Algorithm for generating greedy initial population
This algorithm generates an initial population for genetic algorithm using greedy approach.

1: Generate chromosomes equal to the size of initial population using random approach.
2: Generate some chromosomes using nearest neighbor greedy approach and add these chromosomes in the initial population.
3: Calculate fitness of every chromosome in the complete population generated after step2.
4: Select best chromosomes equal to size of initial population from the sorted population generated in step 3.

Greedy Cross Over Operation – In this work a modified greedy cross over operator has been proposed. In this work classical one point cross over operator using greedy approach is used. Cross over operator generates new children from given parents. The procedure to perform greedy cross over is explained in Algorithm-3.

Algorithm-3 : Algorithm for Greedy Cross Over Operator
This algorithm takes 2 parent chromosomes as input and rearranges these parent chromosomes using greedy approach to generate 2 children.

1: Select parent1 to perform cross over and select a cross over point.
2: From starting to cross over point copy cities from parent to child1.
4: After cross over point add remaining cities of parent-1 chromosome into child-1 chromosome using nearest neighbour first greedy approach.
5: Select parent-v2 to perform cross over.
6: From cross over point to end point copy cities from parent-2 into child-2.

7: From start point to cross over point cities from parent-2 to child-2 using greedy nearest neighbour first approach
8: Return child-1 and child-2 as newly generated children.

Modified Mutation – In mutation small accidental changes has been made in the population. Mutation rate determines the amount of accidental changes to be made in the population. Generally mutation rate is kept very low. In this modified mutation operator the mutation rate is taken quite high and kept 30%. The procedure for modified mutation is given in Algorithm-4.

Algorithm-4 : Modified Mutation

This algorithm takes initial population of chromosomes as input, perform modified mutation on it and generate a mutated population for next iteration.

*1: Select chromosomes equal to the mutation count to perform mutation. (Where mutation count = mutation-rate*size-of-population)*
2: For each chromosome selected for mutation in step-2 perform step-3 and step-4.
3: Select two positions of cities pos1 and pos2 randomly in the chromosome.
4: Swap cities at pos1 and pos2 in the chromosome.
5: Calculate fitness of these newly generated chromosomes and add these in population.
6: Sort population by fitness and discard worst chromosomes from this population to generate population for next generation.

The modified Greedy Genetic Algorithm GGA to solve Travelling Salesman Problem is as follows:

Algorithm – 5: Greedy Genetic Algorithm GGA to Solve Travelling Salesman Problem

This algorithm take a TSP problem as input and give optimal solution for that TSP using Greedy Genetic Algorithm GGA.

1: Encode given problem in genetic form.
2: Initialize genetic algorithm parameters such as size of population, cross over rate, mutation rate etc.
3: Generate initial hybrid population of randomly generated chromosomes and greedy chromosomes generated using Algorithm-2.
4: Select parents chromosomes (from current population) to perform cross over.
5: Perform greedy cross over to generate children using Algorithm-3 .
6: Add newly generated children in population.
7: Discard worst chromosomes from population.
8: Perform modified mutation using Algorithm-4.
9: Generate population for next generation.
10: If stopping criteria reached then STOP otherwise go to step-4.

Proposed greedy algorithm to solve travelling salesman problem (Algorithm-5) has been implemented on some standard TSP problems. The next section illustrates the results found after implementation.

III. RESULT ANALYSIS

Proposed algorithm is implemented using JAVA and applied on standard TSP problems such as Eil51 and Att48. It has been implemented using jdk1.7 and NetBeans8.0.2. TSP-Lib is a TSP library available online that provide us standard TSP problems and their optimal solutions known till date. Table-I shows a comparison of results of proposed algorithm and some other existing algorithms such as SWAP_GATSP, OX_SIM, MOC_SIM [10]. Table-I also compare the results of proposed algorithm with results given in TSP LIB library. Table-II is showing comparison of results of proposed algorithm with well-known heuristic algorithms such as PSO, ACO and HPSACO. The results of the proposed algorithm has been compared with the optimal result given in TSP LIB library. The optimal path length for Eil51and Att48 TSP problems are 426 and 33522 respectively in TSP LIB library. The optimal path length of the proposed algorithm for Eil51 and Att48 TSP problems are 415 and 32139. So the proposed algorithm found a path of length 415 as compared to 426 and hence algorithm perform 2.65% better than the best known value. For TSP problem Att48 the proposed algorithm found a path of length 32139 whereas the optimal path length in TSPLIB is 33667. So the proposed algorithm found a path which is 4.75% better than the path given in TSPLIB.

TABLE I
Comparison of results with SWAP_GATSP, OX_SIM, MOC_SIM and proposed GGA algorithm [10]

Problem		**TSP LIB Results**	**SWAP_GA TSP**	**OX_SIM**	**MOC_SIM**	**[10]**	**Proposed Algorithm GGA**
Eil51 N=51	Best	426	439	493	444	429	415
	Average		442	540	453	434	422
Att48 N=48	Best	33522	-	-	-	-	32139
	Average		-	-	-	-	32403

TABLE II
Comparison of results found with PSO, ACO and HPSACO [11]

TSP Problem	**TSP LIB Results**	**PSO**		**ACO**		**HPSACO**		**Proposed Algorithm GGA**	
		Best	**Average**	**Best**	**Average**	**Best**	**Average**	**Best**	**Average**
Att48	33522	33734	33982	33649	33731	33524	33667	32139	32403

Table-III Best results found for Travelling Salesman Problem using proposed algorithm and using some existing algorithms.

TABLE III
TSP path using proposed algorithm and optimal path using existing algorithms

ATT48		**EIL51**		**EIL76**	
Best Known Path using existing algorithms	**Best Known Path using proposed algorithm**	**Best Known Path**	**Best Known Path using proposed algorithm**	**Best Known Path**	**Best Known Path using proposed algorithm**
1	17	1	40	1	21
8	43	22	19	33	47
38	7	8	41	63	48
31	19	26	13	16	29
44	37	31	25	3	45
18	6	28	14	44	27
7	30	3	6	32	52
28	28	36	48	9	34
6	36	35	27	39	46
37	7	20	51	72	8
19	18	2	46	58	35
27	44	29	12	10	7
17	31	21	47	31	53
43	38	16	18	55	14
30	9	50	4	25	19
36	8	34	17	50	54
46	1	30	37	18	13
33	16	9	42	24	57
20	22	49	44	49	15
47	3	10	15	23	5
21	23	39	45	56	37
32	11	33	33	41	20
39	12	45	39	43	70
48	40	15	10	42	60
5	15	44	30	64	71
42	46	42	34	22	36
24	33	40	21	61	69
10	20	19	29	21	61

45 35 4 26 2 29 34 41 16 22 3 23 14 25 13 11 12 15 40 9 -1 EOF	47 21 13 25 14 34 41 29 5 48 39 32 24 10 42 2 26 4 35 45 -1 EOF	41 13 25 14 24 43 7 23 48 6 27 51 46 12 47 18 4 17 37 5 38 11 32 -1 EOF	2 16 50 9 49 5 38 11 32 1 22 20 35 36 3 28 31 8 26 7 23 24 43 -1 EOF	47 36 69 71 60 70 20 37 5 15 57 13 54 19 14 59 66 65 38 11 53 7 35 8 46 34 52 27 45 29 48 30 4 75 76 67 26 12 40 17 51 6 68 2 74 28 62 73 -1 EOF	62 73 1 33 63 16 51 6 68 75 76 67 4 30 2 74 28 22 64 42 43 41 56 23 49 24 18 50 32 44 3 40 17 26 12 58 72 39 9 25 55 31 10 38 65 66 11 59 -1 EOF
Best Path Length = 33522	Best Path Length = 32139	Best Path Length = 426	Best Path Length = 415	Best Path Length = 538	Best Path Length = 556

Complexity of the proposed algorithm- TSP problem has many application areas in science and engineering. The proposed algorithm goes deeper in the search space and found solutions which are better than the solutions known by other algorithms. The proposed changes in the genetic algorithm increase the complexity of the standard genetic algorithm. But the changes search more and more deep in the search space and found better solutions.

Utility of the Research - This research is useful in the application areas of TSP where a solution for a given TSP problem has to be found once and then it is applied for a long time. As the complexity of the proposed algorithm is more as compared to other existing algorithm so it take more time and need more memory space to find solutions thus the proposed algorithm is not applicable and useful where the algorithm have to be applied

again and again to solve a given TSP. For example drilling electronic circuit is a TSP problem and drilling is done by a robot arm on fixed points. If we apply the proposed algorithm and found a solution for drilling a given type of circuit boards then the solution can be used to drill all the circuit boards of that type and no need to apply the GA to find solution again and again. So the proposed algorithm is very useful in that case.

IV. CONCLUSION

Genetic Algorithm can be used to solve NP-Complete problems such as Travelling Salesman Problem. The performance of the Genetic Algorithm depends upon the performance of its operators which are selection, reproduction (cross over) and mutation. In this paper a solution of TSP has been given by using Genetic Algorithm. In this work the initial population is generated by using a hybrid of random population generation and greedy population generation. Also the cross over operation is a greedy rearrange operator. Mutation rate is taken quite high. By applying these changes the Genetic Algorithm works better and generate solutions which are better than the optimal solutions known by using other algorithms. In future the algorithm can be applied and tested on some other TSP problems on which it has not been applied yet(Specially on large TSPs having more than 500 nodes in the graph). Further greedy approach can also be applied on some other NP-Complete problems such as Graph Coloring Problem, Set Cover Problem etc.

REFERENCES

[1] Mei Mi, Xue Huifeng, Zhong Ming, Gu Yu. An Improved Differential Evolution Algorithm for TSP Problem. In: International Conference on Intelligent Computation Technology and Automation ICICTA; IEEE; 11-12 May 2010; Changsha, China. 544 - 547
[2] Shakeel Arshad, Shengxiang Yang. A Hybrid Genetic Algorithm and Inver Over Approach for the Travelling Salesman Problem. In: IEEE Congress on Evolutionary Computation CEC; 18-23 July; 2010. 1-8.
[3] Oliviu Matei, Petric˘a Pop. An Efficient Genetic Algorithm for Solving the Generalized Traveling Salesman Problem. In: 6th International Conference on Intelligent Computer Communication and Processing ICCP; 26-28 August; IEEE 2010. 87-92
[4] Ren Shuai, Wang Jing, Xuejun Zhang. Research on Chaos Partheno - Genetic Algorithm for TSP. In: International Conference on Computer Application and System Modeling ICCASM 2010; 22-24 October; IEEE; 2010. V1-290 - V1-293
[5] Yupei Xiong, Bruce Golden, Edward Wasil. A One-Parameter Genetic Algorithm for the Minimum Labeling Spanning Tree Problem. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION 2005, FEBRUARY VOL. 9, NO. 1
[6] H.H. Yang, M. -T. Wang, Y. J. Chen, Y. Huang, C.J. Kao. Crossover Based on Rough Sets - a Case of Multidimensional Knapsack Problem. In: International Conference on Industrial Engineering and Engineering Management IEEM; 7-10 December 2010 ; IEEE. 2411 - 2415
[7] Abdelhakim Gharib, Jamal Benhra and Mohsine Chaouqi. A Performance Comparison of PSO and GA Applied to TSP. International Journal of Computer Applications IJCA; 2015, November; Volume 130 – No.15, 34-39
[8] Yang Yi, Qian-sheng Fang. The Improved Hybrid Genetic Algorithm for Solving TSP Based on Handel-C. In: International Conference on Advanced Computer Theory and Engineering (ICACTE); 20-22 August 2010;Chengdu China . IEEE; 2010. Pages: V3-330 - V3-333
[9] Varshika Dwivedi, Taruna Chauhan, Sanu Saxena and Princie Agrawal.Travelling Salesman Problem using Genetic Algorithm. International Journal of Computer Applications (IJCA) Proceedings on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI 2012) DRISTI, 2012. 25-30
[10] Gohar Vahdati, Sima Yaghoubian Ghouchani and Mahdi Yaghoobi. A hybrid Search Algorithm with Hopfield Neural Network and Genetic Algorithm for Solving Traveling Salesman Problem. In: The 2nd International Conference on Computer and Automation Engineering ICCAE; 22-28 February 2010 ;Singapore ; IEEE 2010 . 435 - 439
[11] Hao Jia. A Novel Hybrid Optimization Algorithm and its Application in Solving Complex Problem. International Journal of Hybrid Information Technology IJHIT; 2015 volume(issue): Vol.8, No.2 1-10
[12] ABDOUN Otman, ABOUCHABAKA Jaafar . A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem. International Journal of Computer Applications IJCA; 2011, October; Volume 31– No.11 49-57
[13] Vinod Jain, J S Prasad,An Optimized Algorithm for Solving Travelling Salesman Problem Using Greedy Cross Over Operator, published in 10th INDIACom; INDIACom-2016; IEEE Conference ID: 37465, 2016 3rd International Conference on “Computing for Sustainable Global Development”, 16th - 18th March, 2016.

AUTHOR PROFILE

VINOD JAIN Education Master of Technology(Computer Engineering) YMCA University, Faridabad, Haryana, India(2012), Master of Computer Application MCA, Kurukshetra University (2004), Research Scholar MVN University Palwal, Haryana, India. Currently working as a Assistant Professor, B.S.Anangpuria Institute of Technology and Management Faridabad, Haryana since 2008. He has published more than 8 papers international journals and international conferences. His area of research include Genetic Algorithms, NP-Complete and NP-Hard problems, Search Engine Optimization, Page ranking, Crawling, Indexing, Web mining etc.

JAY SHANKAR PRASAD research interest is Artificial Intelligence, Pattern recognition, Machine learning, Computer Vision, Robotics, Humanoid Robots, Gesture Recognition, ISL Recognition, Pattern mining, Cloud computing etc .He has published Twelve papers in International journals and International conferences. He has 16 years of teaching and 3 years of software industry experience. He also guided many postgraduate and under graduate level projects.