A NEW HYBRID APPROACH FOR SOLVING TRAVELLING SALESMAN PROBLEM USING ORDERED CROSS OVER 1(OX1) AND GREEDY APPROACH

Komal Joshi[1], Ram Lal Yadav[2]

[1]*M.Tech Scholar, Department of Computer Science and Engineering, Kautilya Institute of Technology and Engineering, Jaipur, Rajasthan*

[2]*Associate Professor, Department of Computer Science and Engineering, Kautilya Institute of Technology and Engineering, Jaipur, Rajasthan*

Abstract

Travelling Salesman Problem is a well known NP problem. It is an optimization problem. Genetic Algorithms are the evolution techniques to solve optimization problems. In this paper a new hybrid technique using ordered cross over 1 (OX1) and greedy approach has been proposed. Experiment results shows that the proposed hybrid cross over is better than the existing cross over operator as the new operator provide a better path when executed for the same number of iterations.

Keywords:- *Travelling Salesman Problem, ordered cross over 1 (OX1)*

----------***----------

1. INTRODUCTION

The travelling salesman problem has been first studied in 1800's by Sir William Rowan, Hamilton and Thomas Penyngton Kirkman. Hamilton was an Irish mathematician and Kirkman was a British mathematician. In 1857, Hamilton created a game in which a player has to connect 20 specified points by specified connectors. The game was named as Icosian [6] . Later Hamilton has done research on graph theory and contributes to graph theory. In travelling salesman problem a salesman has to visit a number of cities. Salesman has to start its journey from any of the city called the starting city, cover all the cities in the tour and return back on the starting city. Problem is to find the tour such that salesman have to travel minimum distance to cover all the cities. In 1972 the travelling salesman problem has been declared as NP-complete. NP-complete is a class of problems which are non-deterministic polynomial time hard i.e. no polynomial time algorithm is available to solve these problems. TSP is represented by a Hamiltonian cycle which is a NP-complete class problem. So it is very difficult to find optimal tours to solve this problem. Travelling salesman problem has its application in many areas such as Drilling of printed circuit boards, 2. Overhauling gas turbine engines, X-Ray crystallography, Computer wiring, order-picking problem in warehouses etc.

Genetic Algorithm (GA) is an optimization techniques which uses special operators such as selection, reproduction and mutation to solve problems which are difficult to solve by using traditional techniques. GA works on some optimization function which may be a minimization function or a maximization function. Genetic Algorithm is also applicable in solving optimization problem such as Travelling Salesman Problem. In literature work has been done on many techniques to solve TSP using Genetic Algorithms. A survey of literature on solving TSP using GA is given in the next section.

2. RELATED WORK

Yang Yi and Qian-Sheng Fang [1] proposed an improved Hybrid Genetic algorithm for solving travelling salesman problem on Handel-C. Authors used a greedy approach to improve the performance of genetic algorithm. [5] Poonam Panwar and Sonali Gupta present a survey of soft computing techniques used for optimizing travelling salesman problem. Authors explain that soft computing techniques such as genetic algorithm can be used to solve many problems which belong to NP-complete or NP-hard set of problems. In the paper it has been concluded that soft computing techniques can be applied to solve travelling salesman problem. Author [5] concluded that genetic algorithm perform better in solving travelling salesman problem. The main advantage of GA is that it converges to the solution in very short time. In future work author concluded that a hybrid of existing solutions can be used to solve TSP to enhance the performance of GA. [4] GoharVahdatiet.el.publish a new approach to solve traveling salesman problem using genetic algorithmbased on heuristic crossover and mutation operator. In this paper author proposes a new cross over and mutation technique to solve TSP using GA. Author implemented their work and concluded that the new proposed cross over and mutation operators work better than some existing cross over operators such as OX1, MOC and SWAP operators. [2] Varshika Dwivediet.el. proposed a new strategy to find nearly optimized solution to travelling salesman problem using new cross over technique.

3. PROPOSED WORK

While studying the literature it has been observed that Travelling Salesman Problem is an optimization problem that can be solved using Genetic Algorithms. Selection, cross over and mutation are the important genetic operators in genetic algorithms. In this work a new hybrid cross over operator is proposed to generate new children from the parents. The hybrid operator combines the classical OX1 cross over operator and greedy approach to generate new children. The working of two operators is as follows:

3.1 Ordered Cross Over -1 (OX1)

Ordered cross over OX1 is a cross over operator used to generate new children from two selected parents. To represent travelling salesman problem, permutation encoding is used to represent chromosomes of the population. In permutation encoding every chromosome is a new permutation of all the cities that differ in the order in which cities are visited by the salesman. Here in this example there are 20 cities that are to be visited by the salesman and city-1 is the starting and ending city for the journey. Let two parent,parent-1 and parent-2 that will participate in the ordered cross over as follows:

Parent 1 : 1 13 3 15 8 14 20 12 19 6 7 16 4 18 11 17 5 2 10 9 1

Parent 2 : 1 18 3 19 9 8 13 10 5 17 4 2 12 6 20 16 11 14 15 7 1

Let cross over point is 11th city that is city-7 in parent-1.

Child 1 :1 13 3 15 8 14 20 12 19 6 7 18 9 10 5 17 4 2 16 11 1

Child2 : 1 13 3 8 19 4 18 17 5 10 9 2 12 6 20 16 11 14 15 7 1

3.2 Generation of Child-1 using OX1 Cross Over Operator

To generate child -1 all the cities from parent-1 are copied into child-1 up to cross over point i.e. up to city-7.
Child 1 :1 13 3 15 8 14 20 12 19 6 7

Then remaining cities are copied in the order in which they appear in parent-2. As in parent-2 city-18 is traversed first among all the remaining cities then city-18 will appear after city-7 in the child-1. After city-18 city-9 is traversed first among all the remaining cities so city-9 will come after city-18 in child-1. This process is repeated till all the remaining cities are not added in child-1.

3.3 New Hybrid Greedy Operator

In greedy operation, a chromosome is rearranged according to nearest neighbor first greedy strategy. The distances among the cities are stored in a symmetric matrix. The (i,j)th cell in the matrix represent the distance between ith and jth city in the given travelling salesman problem. The distance matrix is generated randomly. The distance matrix is shown in Figure – 1

Let two paths to perform the greedy operation are as follows

Path – 1:1 8 9 11 13 14 12 7 18 16 4 2 20 5 17 19 6 10 15 3 1

Path – 2:1 2 3 18 5 6 16 9 10 14 12 20 13 4 7 8 17 15 19 11 1

Let cross over point is 11th city that is city-7 in parent-1.

Child-1: 1 8 9 11 13 14 12 7 18 16 4 2 6 20 17 5 10 3 15 19 1

Child-2: 1 16 9 2 6 3 10 14 18 5 12 20 13 4 7 8 17 15 19 11 1

3.4 Generation of Greedy Child-1

To generate greedy child-1, All the cities from start city to cross over point (i.e. from city-1 to city-4) are copied as it is from parent-1 into child-1.

Child-1: 1 8 9 11 13 14 12 7 18 16 4

Then among the remaining cities a city is selected which is nearest to current city-4. City-2 is the nearest city (Refer distance matrix) from city-4 among remaining cities (2 6 20 17 5 10 3 15 19). So the next city after-city-4 is city-2 in child-1. Then current city is city-2 and a city which is nearest among all the remaining cities (6 20 17 5 10 3 15 19) is selected. Now city-6 is nearest city to and is added in child-1. This process is repeated till all the remaining cities are not added in child-1. The generated greedy child using this procedure is

Child-1: 1 8 9 11 13 14 12 7 18 16 4 2 6 20 17 5 10 3 15 19 1

Distance Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1:	0	39	36	46	45	40	26	26	36	23	16	22	13	44	38	20	37	30	37	18
2:	39	0	30	27	26	19	14	27	12	33	19	28	33	40	33	47	39	25	34	33
3:	36	30	0	44	30	19	24	25	28	10	15	33	23	38	30	48	31	47	32	16
4:	46	27	44	0	28	40	42	20	25	47	47	15	36	13	34	37	48	33	33	47
5:	45	26	30	28	0	38	44	37	29	16	10	11	49	36	31	43	20	37	25	23
6:	40	19	19	40	38	0	17	16	19	26	21	35	46	35	12	48	47	24	43	12
7:	26	14	24	42	44	17	0	15	32	30	38	16	36	19	15	19	18	32	49	32
8:	26	27	25	20	37	16	15	0	35	43	41	29	24	15	17	43	29	48	16	46
9:	36	12	28	25	29	19	32	35	0	36	20	46	35	35	26	19	35	14	16	25
10:	23	33	10	47	16	26	30	43	36	0	36	19	26	11	27	19	31	45	15	21
11:	16	19	15	47	10	21	38	41	20	36	0	26	31	29	15	14	25	48	39	39
12:	22	28	33	15	11	35	16	29	46	19	26	0	36	35	14	27	38	37	15	10
13:	13	33	23	36	49	46	36	24	35	26	31	36	0	36	25	46	38	41	16	23
14:	44	40	38	13	36	35	19	15	35	11	29	35	36	0	16	28	39	15	12	14
15:	38	33	30	34	31	12	15	17	26	27	15	14	25	16	0	48	30	44	14	31
16:	20	47	48	37	43	48	19	43	19	19	14	27	46	28	48	0	20	42	19	20
17:	37	39	31	48	20	47	18	29	35	31	25	38	38	39	30	20	0	48	29	15
18:	30	25	47	33	37	24	32	48	14	45	48	37	41	15	44	42	48	0	27	36
19:	37	34	32	33	25	43	49	16	16	15	39	15	16	12	14	19	29	27	0	22
20:	18	33	16	47	23	12	32	46	25	21	39	10	23	14	31	20	15	36	22	0

Fig1- Distance matrix to store distances between all the cities in TSP.

The proposed algorithm to generate child using greedy approach is as follows :

Algorithm: Generate-Greedy-Child (Parent-1,Parent-2,Child-1,cross_over_point)
Input : Parent-1,Parent-2, cross_over_point
Output : Child-1
Step-1 : Select a cross_over_point as the position of middle city in parent-1.
Step-1 : Copy all cities from first city to cross_over_pointfrom parent-1 into child-1.
Step-2 : Add all cities from cross_over_point to last city into list_of_ remaining_cities.
Step-3 : Select next_city from list_of_remianing_cities that is at minimum distance from current city. Add the next_city in child-1 and remove the next_city from list_of_remaining_cities.
Step-4 : Repeat step-3 until list_of_remianing_cities is not empty.
Step-5 : Return child-1.

The above algorithm is used to generate child using greedy approach of nearest neighbor first. The proposed genetic algorithm using a hybrid of OX1 cross over and greedy operator is as follows:

Algorithm : Genetic Algorithm for solving travelling salesman problem using hybrid of Ordered Cross Over -1 (OX1) and greedy cross over operator.

Step-1: Generate a random initial_population ofchromosomes using permutation encoding.
Step-2: Calculate fitness (Path Length) of every chromosome of initial_population.
Step-3: Repeat step-4 to step-7 for given number of generations (iterations).
Step-4: Select parents from initial_populationusing random selection for cross over.
Step-5: Perform ordered cross over OX1 and Greedy_Cross_Over to generate children from parents.
Step-6: Calculate fitness of all newly generated children of step 4. Add newly generated children in initial_population.
Step-7: Sort combined population initial_population + new_children by path length and select best chromosomes for next_population. Set initial_population = next_population and go to step-4.
Step-8: Stop

4. RESULTS AND ANALYSIS

The proposed algorithms has been implemented for the given distance matrix. After implementing the classical OX1 operator and new combined (OX1 and greedy) some good results have been found. The results are summarized in Table-1.

Table 1- Path length of best path using existing work and proposed new work

Iteration Count	Path length of best path using OX1 (Existing Work)	Path length of best path using Combined OX1 and Greedy cross over operation (Proposed Work)
1	460	429
2	460	343
3	435	343
4	435	343
5	435	343
6	435	343
7	435	343
8	435	343
9	435	343
10	435	333
20	413	333
30	413	333
40	413	333
50	413	333

It has been observed that path length of best path is 413 using existing OX1 cross over and 333 using proposed new approach. The path length is 80 unit less using proposed work as compared to existing work (413-333=80). So path length of best path is 24% better than the existing work.(80 is 24% of 333). So our work is 24% better than the existing work. Figure 2 shows a graphical comparison between two techniques.

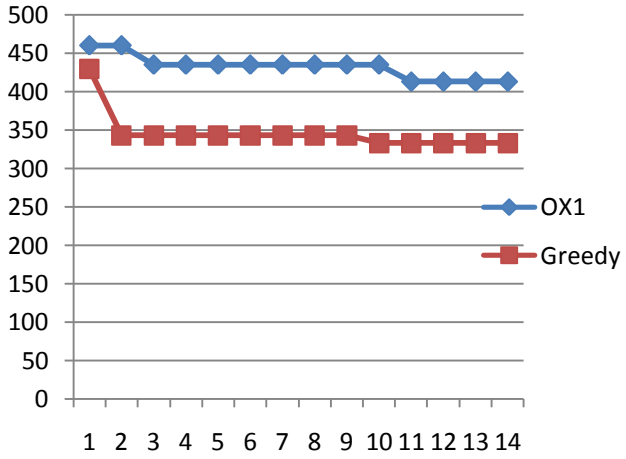

Fig 2 - Comparison of existing work and new proposed work.

5. CONCLUSION

Genetic algorithm (GA) is a relatively new optimization technique based on natural evolution which can be applied to Travelling Salesman Problem. Genetic Algorithm (GA) solves TSP using genetic operators such as selection, mutation and cross over. The performance of genetic algorithms mainly depends upon the performance of genetic operators. Cross over is a very important operator that improve the convergence of GA towards solution. In this work a hybrid of existing OX1 cross over operator and greedy cross over operator such as nearest neighbor first is applied to solve TSP. After analyzing the results the proposed hybrid cross over operator gives a path with a path length 20-30 percent smaller.

FUTURE WORK

However the proposed algorithm is working better than the existing work but it can be implemented and tested on some standard TSP problems such as Oliver30, eil51, eil76, KroA150 etc. Further algorithm can be tested in future for various values of genetic parameters such as size of population, cross over rate, mutation rate and their effect on the performance of the Genetic Algorithm can be tested. The work can be compared on many other cross over operators such as OX2, PMX etc.

REFERENCES

[1] VarshikaDwivediet.el. "Travelling Salesman Problem using Genetic Algorithm", National Conference on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI) 2012. Proceedings published in International Journal of Computer Applications® (IJCA). Pg 25-30.

[2] Yang yi and Qian-sheng Fang, "The Improved Hybrid Genetic Algorithm for Solving TSP Based on Handel-C", 20IO 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 978-1-4244-6542-2/$26.00 © 2010 IEEE

[3] Z. Tao, "TSP Problem Solution Based on Improved Genetic Algorithm", In Proceedings of the 2008 Fourth international Conference on Natural Computation, ICNC. IEEE Computer Society, Washington, DC, vol. 01, pp. 686-690, 2008.

[4] GoharVahdatiet.el. , "A New Approach to Solve Traveling Salesman Problem Using Genetic AlgorithmBased on Heuristic Crossover and Mutation Operator", 2009 International Conference of Soft Computing and Pattern Recognition, 978-0-7695-3879-2/09 $26.00 2009 IEEE.

[5] PoonamPanwar and SaloniGupta , "Brief Survey of Soft Computing Techniques Used for Optimization of TSP". The International Journal of Computer Science &Applications (TIJCSA), Volume 2, No. 01, March 2013 ISSN – 2278-1080.

[6] Cook, William. "History of the TSP."The Traveling Salesman Problem. Oct 2009. Georgia Tech, 22 Jan 2010. <http://www.tsp.gatech.edu/index.html>.

[7] Calgor,H."TheBrute Force Algorithm".24 January 2010#THE BRUTE FORCE ALGORITHM <http://www.ctl.ua.edu/math103/hamilton/HCalgor.htm >.