IJCSE International Journal of Computer Sciences and Engineering **Open Access**

Research Paper **Volume-5, Issue-8** **E-ISSN: 2347-2693**

A Hybrid Method for Solving Traveling Salesman Problem using Hungarian Method

A.K.Prasad[1*], Pankaj[2]

[1*]DST-CIMS, Banaras Hindu University, Varanasi, India
[2]Mathematics Section, Mahila Maha Vidyalaya, Banaras Hindu University, Varanasi, India

[*]Corresponding Author: anubhavkrprasad@gmail.com

Available online at: www.ijcseonline.org

Received: 08/Jul/2017, Revised: 17/Jul/2017, Accepted: 20/Aug/2017, Published: 30/Aug/2017

Abstract—Genetic Algorithms are earning respect in different fields of Operation Research like Transportation and Traveling Salesman Problem, etc. However, the best solution they produce needs several iterations to obtain. This paper develops a new hybrid method for Travelling Salesman Problem. The proposed hybrid method delivers same solution each time unlike genetic algorithms. The efficiency is compared against following existing crossover operators; namely, Order Crossover, Modified Order Crossover, Sequential Constructive Crossover, Modified Sequential Constructive Crossover. Experimental results show that the proposed hybrid method is better than the compared methods.

Keywords— Travelling salesman problem, NP-hard, Hybrid method, Genetic algorithm, Hungarian method.

I. INTRODUCTION

Traveling Salesman Problem

Combinatorial optimization problem [**1**] contains a broad study of the Travelling Salesman Problem (TSP) [**2**]. The problem is to obtain an optimized route for a salesman who needs to visit each city exactly once and returns to its starting city such that the cost of its entire tour is minimized to an optimum level. The problem can be well represented by a weighted graph G having ***N*** cities and ***E*** paths between cities. The graph is complete graph with each city connected to another city with edge e. Let G = (N, E) be a graph where N is a set of vertices representing cities and ***E*** is the set of edges representing paths. Let c_{ij} be a distance between each i^{th} and j^{th} city having edge e ϵ ***E***, which can be calculated using following Euclidean distance:

$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

where (x_i, y_i) and (x_j, y_j) represents coordinates of city i and city j.

Above formulation can be represented using matrix, known as cost matrix. $C = (c_{ij})$, i, j=1, 2, 3,…,m, where m is the total number of distinct paths formed for each city. In cost matrix C the $(i, j)^{th}$ entry, c_{ij}, represents the cost of travel from i^{th} city to j^{th} city. The path with least c_{ij} is considered as shortest path.

Literature Review

The importance of the Travelling Salesman Problem (TSP) is that it belongs to a larger class of problems known as combinatorial optimization problems. TSP is broadly recognized as NP-hard problems **[3]** as no algorithm is able to solve it in the polynomial time. **Hamiltonian cycle** [**4**] and TSP both, ask for a tour to visit each vertex exactly once. However, TSP works on weighted graphs, while Hamiltonian cycle works on un-weighted graphs. The nominal anticipated time to get optimal solution is exponential. The problem is to find the shortest possible route that a salesman can follow in-order to visit each city exactly once and return to the starting point/city. The problem seems to be pretty easy to solve, but time expands exponentially as there is no short-cut way of finding the optimal route in terms of distance/time. It is pretty obvious that techniques like Brute-force could not be applied to such problems as the time will depend on the combination of given number of cities which is not feasible. For instance, consider a typical case of 16 cities which will have 653,837,184,000 distinct routes**.** However, different algorithms have been developed which are able to produce optimized routes or nearly close to it using Heuristics and Exact algorithms. Cutting plane or Branch and Bound method **[5, 6]** is one of the most effective exact algorithms, but has high time complexity. On the other hand, a heuristic or meta-heuristic algorithm takes considerable less time than exact algorithms and is able to provide near optimal solution. The most commonly used heuristic and meta-heuristic algorithms are Greedy algorithms, 2-opt algorithms, Simulated Annealing [**7**], Tabu Search [**8**], Ant Colony

Optimization [**9**], Genetic Algorithms [**10**], etc. In this paper, Modified Sequential constructive crossover (MSCX) **[11]** was compared with the Proposed Hybrid method which in turn is compared with three other Genetic algorithms namely Order Crossover (OX) [**12**], Modified OX (MOX) [**13**] and Sequential Constructive Crossover (SCX) [**14**] respectively. A complete historical development of TSP and related problems can be found in Hoffman and Wolfe [**15**], Applegate et al. [**16**] and Cook [**17**].

Rest of the paper is organized as follows, Section I contains the introduction of Traveling Salesman Problem , Section II contain the related work of compared methods: OX, MOX, SCX, MSCX, Section III describes the Proposed Hybrid Method, Section IV contain the algorithm for Proposed Hybrid Method, Section V describes results and conclusion in Section VI.

II. Related Work

Order Crossover (OX)

OX starts by selecting two random cut points on parent chromosomes. In order to get the offspring, the string between two randomly selected cut points of parent1is copied to the same position of offspring first, then the remaining positions of the offspring is filled by considering the sequence of cities in parent2 starting from the second cut point and continuing till all the positions of the offspring is filled from right to left. In figure 1, the randomly selected cut points is from position 4 to 6, the substring 467 of parent 1 is copied to the offspring in the same position as in parent 1 (step1). The remaining positions in the offspring is filled by starting from position next to the last filled position in offspring by copying the strings of parent2 aftersecond cut point of parent2, i.e. 36154728 (step2) such that there is no repetition of cities in offspring. Hence, city 6 is not copied from parent2 as it is present in offspring from parent1. Same condition is with cities 4 and 7.

Parent1 : 1 3 8 **4 6 7** 2 5

Parent2 : 1 5 4 **7 2 8** 3 6

Offspring : - - - **4 6 7**- - - (Step1)
5 2 8 **4 6 7** 3 1 (Step2)

Figure1.OrderCrossover (OX)

Modified Order Crossover

OX starts by selecting two random cut points on parent chromosomes. In order to get the offspring, the string between two randomly selected cut points of parent1is copied to the same position of offspring first, then the remaining positions of the offspring is filled by considering the sequence of cities in parent2 starting from the second cut point and continuing till all the positions of the offspring is filled from right to left. In figure 1, the randomly selected cut points is from position 4 to 6, the substring 467 of parent 1 is copied to the offspring in the same position as in parent 1 (step1). The remaining positions in the offspring is filled by starting from position next to the last filled position in offspring by copying the strings of parent2 after second cut point of parent2, i.e. 36154728 (step2) such that there is no repetition of cities in offspring. Hence, city 6 is not copied from parent2 as it is present in offspring from parent1. Same condition is with cities 4 and 7.

Parent1 : 1 3 8 **4 6 7** 2 5

Parent2 : 1 5 4 **7 2 8** 3 6

Offspring : - - - **4 6 7**- - - (Step1)
5 2 8 **4 6 7** 3 1 (Step2) After Applying Order Crossover (OX)
5 2 8 **7 6 4** 3 1 (Step3) After Applying Modified Order Crossover (MOX)

Figure2. Modified Order Crossover (MOX)

Sequential Constructive Crossover (SCX)

SCX starts by selecting two parent chromosomes from the matting pool. In order to get the offspring, a point, called crossover site, along common length of two selected parents is randomly selected and the information after the crossover site of the two parent strings are swapped creating two new children. SCX operator constructs an offspring using better edges on the basis of their values present in the parents' structure. It also uses the better edges, which is missing in the parent's structure. Hence, the chances of producing a better offspring are more than those of Edge Recombination Crossover (ERX) [**18**] and N-point crossover (GNX) [**19**]. Figure3 shows the process for SCX. The starting node is 1 of parent1which is copied to the offspring. From node 1, node 3 and node 5 are available from parent1 and parent2 respectively, but the least distance is between node 1 and 3, hence node 3 is copied to the offspring. Similarly node 6 is copied to the offspring. Now, the available unvisited node from node 6 is node 7 in parent1 with distance 25 and for parent2, node 6 is connected to node 1 which is present in offspring, hence the node for parent2 is to be selected from the set (2,3,4,5,6,7,8) which is node 2. The distance is smaller between node 6 and 7 with distance 25 (for node 6 and 2 it is greater than 25), hence node 7 is copied to offspring. The process continues to have the offspring with path 13672548.

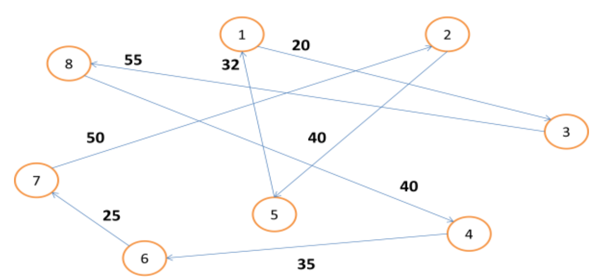

Fig.3(a). Parent1 for SCX Crossover (13846725)

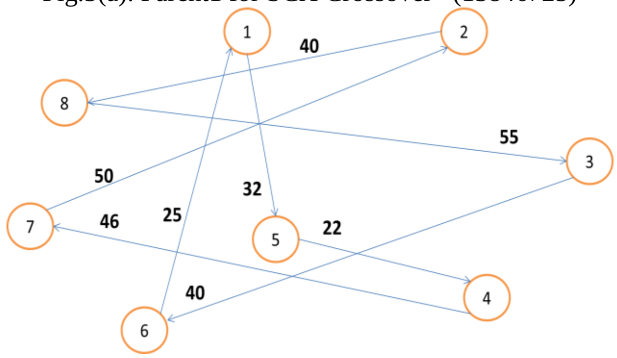

Fig.3(b). Parent2 for SCX Crossover (15472836)

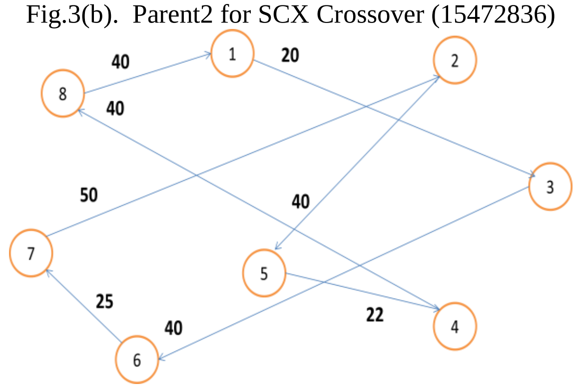

Fig.3(c). Offspring for SCX Crossover (13672548)

Modified Sequential Constructive Crossover (MSCX)

MSCX starts by selecting two parents from the matting pool. In order to get the first offspring, first node from parent1 is copied to the offspring. After that, the first non visited node from both the parents is compared and the node with the least distance is copied to the offspring. The process continues until all the nodes are copied to the offspring.

III. Proposed Hybrid Algorithm

The Proposed hybrid method unlike GA is able to produce the same solution for each execution. It is based on the idea of Hungarian Method [**20**]. It generates ***m*** initial solution, where ***m*** is the number of cities using the basics of Hungarian Method. The initial solution obtained is further refined by inserting sub-paths between cities having maximum distance. This process continues until no further sub-paths could be found. The refined paths obtained contain the desired output with one or more paths giving the same optimized cost.

IV. Algorithm

Step 1: Starting from city ***i***, mark city having least distance from city ***i*** (row wise). (***i***=***1*** to ***N***)
Step 2: Repeat step 1for each city.
Step 3: Construct paths with the marked cities with each city contributing one path using step 3.
Step 3(a): Calculate distance, ***d*** and ***d'*** between city ***i- j*** and city ***j-k*** respectively, where ***j*** and ***k*** is the current shortest distance from city ***i*** and ***j*** respectively.
Step 3(b): Calculate distance, ***d''*** and ***d'''*** between city ***i-j'*** and city ***j'-k'***, where ***j'*** and ***k'*** is the next current shortest distance from city ***I*** and current shortest distance from city ***j'*** respectively.
Step 3(c): If ***d+d'*** < ***d''+d'''***, connect city ***i*** with city ***j*** and city ***j*** with city ***k*** respectively, else connect city ***i*** with city ***j'*** and city ***j'*** with city ***k'***.
Step 4: Insert sub-path between cities having maximum distance for each path formed. Sub-path is the path cut from the same path and inserted between cities having maximum distance.
Step 5: Repeat step 4 until no such sub-path exists which reduces the cost/length between two cities.

V. Results Analysis

Data used for comparison is shown in table 1 [**11**].The result comparison is shown in table 2 which clearly shows that the Proposed Hybrid method outperforms OX, MOX and SCX for cities ranging from 15 to 20. However, the results are same for MSCX for all comparisons accept for 19 cities where Proposed Hybrid method gives better result.

Table1. Distance matrix between 20 cities from dataset

City	X	Y
1	0	3
2	1	5
3	4	5
4	5	2
5	4	0
6	1	0
7	3	7
8	6	2
9	8	4
10	7	6
11	10	0
12	9	2
13	5	3
14	4	6
15	6	1
16	11	2
17	7	4
18	8	3
19	2	9
20	3	10

Number of cities	Short path GA(OX)	Short path GA(MOX)	Short path GA(SCX)	Short path GA(MSCX)	Proposed Hybrid Method
5	15.6344	15.6344	15.6344	15.6344	15.6344
6	16.7967	16.7967	16.7967	16.7967	16.7967
7	18.8612	18.8612	18.8612	18.8612	18.8612
8	20.3045	20.3045	20.3045	20.3045	20.3045
9	23.6504	23.6504	23.6504	23.6504	23.6504
10	24.9257	24.9257	24.9257	24.9257	24.9257
15	34.5500	33.9444	37.9843	33.2849	33.2849
16	36.9362	35.944	39.095	35.285	35.285
17	37.825	37.1805	37.1805	36.0488	36.0488
18	38.226	38.048	38.6543	36.2269	36.2269
19	42.3098	42.1952	42.6673	41.6533	39.7577
20	44.3428	42.4478	43.8575	41.9358	41.9358

Table2. Short paths by using GA with OX, MOX, SCX, MSCX and Proposed Method for different cities

III. Conclusion And Future Scope

This paper proposed Hybrid method which is based on Hungarian method to solve TSP with sub-paths being added to reduce city distances. The results show that for fewer numbers of cities (5 to 10), all the methods have the same result. However, as the size grows, the performance of OX, MOX and SCX decreases but MSCX and Proposed Hybrid method gives the same result. However, Proposed Hybrid method outperforms MSCX for 19 cities data-set.
Here, the dataset taken is comparatively smaller in size.
However, this technique can be applied to higher datasets.

References

[1] B.D. Jackson and D. Thoro, "*Applied Combinatorics with Problem Solving*", Addison-Wesley Publisher, pp. 159-163, 1990**.**

[2] .E.L. Lawler, J.K. Lenstra, A.R. Kan, D.B. Shmoys, "*The traveling salesman problem: a guided tour of combinatorial optimization*", Wiley, 1985.

[3] Arora, Sanjeev, "Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems", Journal of the ACM (JACM), Vol. 45, Issue.5, pp.753-782, 1998.

[4] A. Tucker, "*Applied Combinatorics*", Sixth Edition, John Wiley & Sons, Inc. 2012.

[5] E. Balas and P. Toth, "*Branch and bound methods. In E.L. Lawler et. al, editor, The Traveling Salesman Problem*", John Wiley & Sons, Essex, pp. 361-401, 1985.

[6] E. L. Lawler and D. E. Wood, "*Branch-and-bound methods: A survey. Operations Research*", Vol. 14, pp. 699-719, 1966.

[7] S. Kirkpatrick, C. Gellat, M. Vecchi, "*Optimization by simulated annealing, Science*", pp. 671–680, 1983.

[8] F. Glover, "*Tabu search, part I*". ORSA J. Computing Vol. 1, pp. 190–206, 1989.

[9] M. Dorigo, C. Blum, "*Ant colony optimization theory: a survey*", Theor. Comput. Sci, Vol. 344, pp. 243–278, 2005.

[10] D. Goldberg, "*Genetic algorithms in search, ptimization, and machine learning*", Addison-Wesley Professional,Boston, 1989.

[11] S.M.A. Moetty, A.O.Heakil, "*Enhanced Traveling Salesman Problem Solving using Genetic Algorithm Technique with modified Sequential Constructive Crossover Operator*", International Journal of Computer Science and Network Security Vol. 12, Issue. 6, pp. 134, 2012.

[12] I.M. Oliver, D.J. Smith and J.R.C. Holland, "*Study of permutation crossover operators on the travelling salesman problem*", In the proceedings of the 1987 Second International Conference on Genetic Algorithms, Cambridge, MA, pp. 224-230, 1987.

[13] A. Rani, S.K. Boora, "*A Novel Commence For Optimizing Task Scheduling In Heterogeneous Multiprocessor Environment Using Genetic Algorithm*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.7, pp.51-56, 2014..

[14] Z.H. Ahmed, "*Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator*", International Journal of Biometrics & Bioinformatics, Vol.3, Issue. 6, pp.96-105, 2010.

[15] Michael Jünger, Gerhard Reinelt,Giovanni Rinaldi, " *The traveling salesman problem*",Handbooks in Operations Research and Management Science Vol.7, pp. 225-330, 1995.

[16] D. Applegate, R.E. Bixby, V. Chvátal, and W. Cook, "*The Traveling Salesman Problem: A Computational Study*", Princeton University Press Publisher, New Jersy, pp. 81-125, 2006.

[17] W. Cook, "*In Pursuit of the Salesman: Mathematics at the Limits of Computation*", Princeton University Press Publisher, Princeton, USA, pp. 19-92, 2011.

[18] 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 Publisher, New York, pp. 350-372, 1991.

[19] N.J. Radcliffe and P.D. Surry, "*Fitness Variance of Formae and Performance Prediction*" , Appears in "Foundations of Genetic Algorithms III", Ed: L.D. Whitley, M.D. Vose, Morgan Kaufmann (San Mateo), pp. 51-72, 1994.

[20] H.W. Kuhn, "*The Hungarian Method for the Assignment Problem*", Naval Research Logistics, Vol. 2, pp.83-97, 1955.

Authors Profile

Mr.Anubhav Kumar Prasad pursed Master of Computer of Computer Applications from School of Management Sciences, India in 2012. He is currently pursuing Ph.D. from Department of Science and Technology-CIMS, BHU, India since 2014. His main research work focuses on Optimization Techniques in Traveling and Transportation Problems, Network Security, Data Structure.

Dr. Pankaj pursed Master of Science (Mathematics) from IIT Roorkee, India. He pursued Ph.D. form BHU, India and currently working as Assistant Professor in Mathematics Section, MMV, BHU, India.. His main research work focuses on Generalised Convex Optimization, Vector Variational Inequalities, Duality.