IJCTA, 10(07), 2017, pp. 245-252

A Metameric Genetic Algorithm with New Operator for Covering Salesman Problem with Full Coverage

Siba Prasada Tripathy* Amit Tulshyan Samarjit Kar*** and Tandra Pal******

Abstract : Post disaster management is a tough and challenging task which carries risks of plenty of lives. To carry on the relief and mass fatality management activities, it is very difficult to reach to all the places of the affected area. Hence, we need Covering Salesman Problem (CSP), a variant of Traveling Salesman Problem (TSP), which can be used to tackle such type of situations. A customer is said to be covered, if it lies within a pre-specified distance of a visited facility on the tour. In this paper, we consider a CSP problem with full coverage and propose a Metameric Genetic Algorithm (MGA) with a new crossover operator. The results are compared with those of various algorithms existing in the literature which shows that the proposed algorithm performs better with respect to cost as well as execution time. Standard TSP instances are considered for evaluating the performance of the CSP, implemented with the proposed algorithm.
Keywords : Traveling Salesman Problem, Covering Salesman Problem, Genetic Algorithm, Metameric structure.

1. INTRODUCTION

The situation after an artificial or a natural disaster becomes exacerbated which lead to heavy demise of lives. At the time of humanitarian relief operation and mass fatality management [1], the rescue or relief team attempts to serve each of the places in the affected area.

Due to limited time or resources, it is difficult for the team to reach every node in a single attempt. Hence, by traversing some of the places [4] and inviting nearby localities to the visited spot is a better way to accomplish the task. A Travelling Salesman Problem (TSP) [2] is to find a least cost Hamiltonian path on a given fully connected graph. There are many variations of TSP [3] considering different aspects and accordingly solved by numerous methods. Covering Salesman Problem is a generalization of TSP [4].

The CSP can be defined as to find the tour with minimum length traversing subset of points (*i.e.*, nodes in a graph) such that every customer which is not on the tour is bounded by a predetermined covering distance from any of the visited nodes, called facilities. The CSP can be applied in mass fatality management, humanitarian relief transportation, telecommunication network [1], [5], etc.

CSP is first introduced in [4], where the goal is to find a Hamiltonian tour with a minimum length visiting a subset of nodes routing in such a way that it maximizes the covering nodes lying within a predefined distance and not in the tour. This work is extended in [6], where the geometric version of the covering salesman problem is demonstrated with bounded error ratio and polynomial time approximation algorithms. Problems in integer programming, mentioned in [7] can be overcome by diving a graph into a number of sub-graphs [7] which is also a variant of the TSP.

* Department of Computer Science and Engineering, NIT Durgapur, India. *E-mail: siba.dilu@gmail.com, siba.dilu@gmail.com*
** Department of Information Technology, NIT Durgapur, India. *E-mail: atulshyan96@gmail.com, atulshyan96@gmail.com*
*** Department of Mathematics, NIT Durgapur, India. *E-mail: dr.samarjitkar@gmail.com, dr.samarjitkar@gmail.com*
**** Department of Computer Science and Engineering, NIT Durgapur, India. *E-mail: tandra.pal@gmail.com*

Many works in the literature introduce different variants of TSP including the pickup-and-delivery [8], multi-depot multiple TSP [9], online TSP [10], clustered TSP [11], generalized TSP [12], [13], etc. Detail of the variants of TSP are presented in [3]. In covering tour problem (CTP) [14], several classes of constraints are demonstrated using an exact branch and cut algorithm with polyhedral properties taking large instances up to 600 vertices. There are three groups of vertices S1, S2 and S3, where all the vertices of group S1 are visited by the tour, those of group S2 are covered by the tour and remaining which are not visited as well as not covered by the tour form the group S3.

The authors in [15] introduces the multiple vehicle cumulative covering tour problem being motivated to solve the problems in humanitarian logistics. The objective is to determine a set of tours that must be followed by a fleet of vehicles in order to minimize the sum of arrival times (latency) at each of the visited locations. There are three types of locations mentioned in the work such as mandatory, optional and unreachable. A variant of TSP in [12] demonstrates cluster optimization neighbourhood, fragment optimization neighbourhood and TSP-inspired neighbourhood in Generalized Travelling Salesman Problem (GTSP) where many disjoint clusters are formed by considering the set of vertices.

There is a variant of CSP proposed in [16], where some natural generalizations of covering salesman problem are proposed. The authors reviewed the relaxation of visiting or covering one vertex only once. Considering some rural health care delivery systems, the authors in [16], considered overnight stay and visiting a place more than once. The authors also proposed two mathematical formulations on CSP based on the node and the flow of the tour where a given amount of nodes must be covered. Also proposed two meta-heuristic algorithms: Memetic Algorithm (MA) and Variable Neighbourhood Search (VNS) which then compared with the exact models.

A hybrid heuristic algorithm has been developed in [17], combining Ant Colony Optimization (ACO) [18] and a dynamic programming approach for CSP where the starting node is a dummy vertex which does not cover any other node and the tour completes by covering a subset of nodes in order to meet the required demand. Recently, in [19], a CSP is considered to maximize the coverage demand within a time limit developing a branch and cut algorithm and assumed that the demand of the vertex is fully covered if it is a part of the tour and partially covered if it is not a part of the tour but close to the traversed vertex.

In this paper, we define CSP with full coverage and solved it with a new metameric genetic algorithm [21] with a new crossover operator.

2. PROBLEM DEFINITION

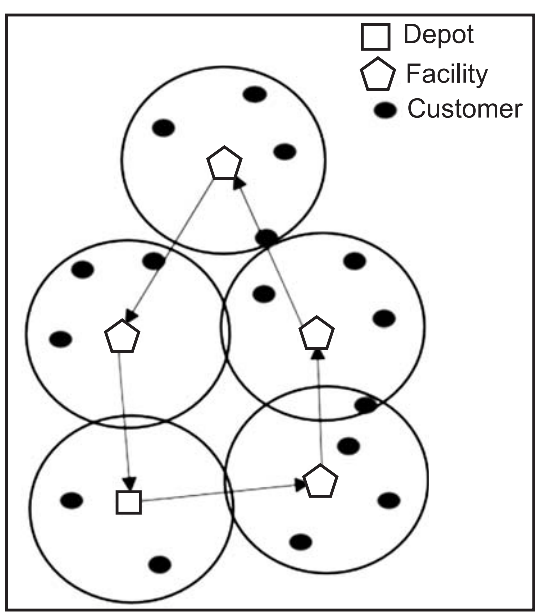

Figure 1: An example of CSP

We consider CSP which resembles to the emergency conditions while handling the dead bodies during mass fatality management, including relief and rescue operations right after various natural or artificial disasters. The problem also includes rural health care delivery for vaccination to prevent epidemic diseases as well as for assuring security in any war like environments. In such type of situations, to complete the task or operation, the team should visit a subset of places covering the pre-specified number of other places as shown in Fig. 1. In Fig. 1, the journey starts from a node called depot, traverses a subset of nodes called facilities and covers some other nodes called customers and comes back to the depot.

3. PROPOSED ALGORITHM (MGA_GPX)

We have developed a modified Metameric Genetic Algorithm (MGA) [21] with a new crossover operator termed as Global Parent Crossover operator (GPX) for CSP, explained below in detail.

Metameric Genetic Algorithm (MGA) is a special kind of genetic algorithm characterized by its chromosomes, consisting of a series of metavariables. Each metavariable G contains two variables. First one represents a facility point F_i and the second one NC_{Fi} represents the corresponding number of nearest customers covered by the facility F_i. The structure of a chromosome is given in Table 1.

Table 1

Metameric form of Chromosome structure

G_0	G_1	– – –	G_n
F_0, NC_{F_0}	F_1, NC_{F_1}	– – –	F_n, NC_{F_n}

MGA deals with the implementation of genetic operators for variable length chromosomes, which differ in their sequence or structure. While moving from one node to another, NC number of nearest nodes or customers are considered to be covered by a node representing a facility F_i. Apart from these NC nodes, next node is chosen from T_N nodes for the next node to be visited. Here T_N is the nearest three nodes which are not in NC nearest nodes. In the algorithm, a gene corresponds to a node or city and a chromosome corresponds to a path, represented by a sequence of nodes. This algorithm can be considered as variable length chromosome based Metameric Genetic Algorithm because of its dynamic nature of the length of the chromosome. The steps involved in modified MGA are explained below.

1. **Setting MGA Parameters and instances :** Parameter setting for MGA is given in Table 3, where T_N represents three nearest customers to be considered while moving from one facility to another. β is the probability of selecting a facility to be removed from the tour in mutation. If improvement has not been occurred continuously for 5 times, the algorithm will be terminated. For this proposed algorithm, P_c is the crossover probability and P_m is the mutation rate. Using all these parameters, the proposed MGA has been implemented and applied to the instances of CSP.
2. **Construct initial population :** We have considered two rules, described below, for initialization of population. 80% chromosomes of the population are generated using Rule 1 and rest 20% are generated using Rule 2.

   Rule 1: After setting the genetic parameters, initialization of population starts. For the next vertex to be visited by the tour it considers T_N consisting of three nearest neighbour out of which the node covering maximum number of customers is selected. The facilities are selected incessantly until the total demand is met.

   Rule 2: This rule states that the next node is chosen randomly out of T_N neighbouring nodes without considering the coverage capacity.
3. **Evaluate Fitness :** The fitness of each chromosome is calculated by adding the lengths of the edges visited by the tour.

4. **Crossover using GPX operator :** A new crossover operator named as GPX operator has been proposed, where we assume a Global Parent (GP) chromosome. GP Chromosome contains all the genes (*i.e.*, nodes) available for a particular instance of the problem. Table 2, given below, shows the GP for the TSP instance eil51. The pseudocode of crossover with GPX operator is given in Algorithm 1.

   In the proposed crossover, the children chromosomes are generated in two phases. In the first phase, we consider two parents, P_1 as parent 1 and P_2 as parent 2, which generate two intermediate children named as IC_1 and IC_2 using single point crossover. Then, IC_1 and IC_2 both take part in crossover separately with the Global Parent (GP) Chromosome. The genes, those are not are yet covered, are copied to the intermediate children considering the genes with its corresponding NC and placed there with minimum insertion cost. Finally two children child1 and child 2 will be generated after crossover. Here, the insertion cost is determined by calculating the total length, keeping the gene at every index position separately in the chromosome. Hence, the time complexity of calculating minimum insertion cost is $O(l)$, where l is the length (variable) of the chromosome. Thus, the time complexity of crossover is $O(mn)$, where m is the population size (P_S) and n is the maximum size of the chromosome in the pool.

Table 2

Global parent chromosome for eil51 instance

0	1	2	…	50

h**Algorithm 1: Pseudocode of crossover with GPX operator**

for iter = 0 to $P_S \times P_c$ // P_S is the population size and P_c is the crossover probability

find two different chromosome from the population randomly, such as P_1 and P_2

$R_p \leftarrow$ a random crossover point where R_p < size of smaller parent

for $i = 0$ **to** $(R_p - 1)$

$IC_1[i] \leftarrow P_1[i]$

end for

for $i = R_p$ **to** $(P_2.\text{length} - 1)$

$IC_1[i] \leftarrow P_2[i]$

end for

remove duplicate gene(*s*) from IC_1 if any

for $i = 0$ to$(R_p - 1)$

$IC_2[i] \leftarrow P_2[i]$

end for

for $i = R_p$ to$(P_1.\text{length} - 1)$

$IC_2[i] \leftarrow P_1[i]$

end for

remove duplicate gene(*s*) from IC_2 if any

while all cities are not covered by child1 do

eCity $\leftarrow$ extract an uncovered city from global parent (GP)

mIndex $\leftarrow$ index with minimum insertion cost

IC_1[mIndex] $\leftarrow$ eCity

end while

Child1 ← IC_1

while all cities are **not** covered by child2 **do**

eCity ← extact an uncovered city from GP

mIndex ← index with minimum insertion cost

IC_2[mIndex] ← eCity

end while

Child2 ← IC_2

end for

5. **Induced mutation :** For mutation, a random facility is extracted from the tour with probability β and other nearest unvisited nodes are substituted with the extracted facility with minimum insertion cost. If it does not give better result, this mutation is discarded and continue the process with mutation rate P_m. Thus, the mutation may reduce the tour length.
6. **Selection :** Two types of selections we have considered: binary tournament selection method and Roulette wheel selection method for selecting the pool for next generation. 50% of population chromosomes are selected by applying binary tournament and remaining 50% are selected by Roulette wheel selection method.

4. COMPUTATION RESULTS AND DISCUSSION

The proposed MGA is implemented with the GPX operator and then applied to 16 standard TSP instances given in [20]. In the standard instances, each customer covers its 7, 9 and 11 nearest customers, resulting in 48 instances. Then, the results of proposed MGA has been compared with two existing methods Current and Schilling [4] and Memetic Algorithm [5] which are shown respectively in Table 4 and Table 5. The Table 4 contains the results of 48 instances where NB and TB respectively represent the number of nodes visited by the tour and time taken by the algorithm for best result.

The algorithm is executed five times and the best and average performance are given in the Table. The algorithm for MGA terminates if there is no improvement for consecutive 5 times. We compared our results with the results of Current and Schilling [4] and we found 37 out of 48 instances giving better result for our proposed MGA_GPX.

The result indicates that the proposed heuristic MGA_GPX outperforms the Current and Schilling [4] heuristic for covering salesman problem where all the customers are covered. The bold texts in the table shows the better results for that corresponding instance.

Table 5 reveals that the proposed algorithm outperforms the Memetic algorithm [5] with respect to its execution time by 25% though it worsens the length of the tour by 14%.

Table 3
Parameters for the proposed MGA

Parameters	*Values*
T_N	3
β	0.5
P_c	0.8
P_m	0.3

Table 4

Comparison of results of CSP using MGA_GPX with the Current and Schilling [4]

Name	*Current and Schilling [4]*				*Proposed MGA_GPX*		
	NC	*Best obj.*	*NB*	*TB*	*Best obj.*	*NB*	*TB*
eil51	7	194	7	0.07	172	9	0.169
	9	169	6	1.92	163	7	**0.307**
	11	167	5	0.59	155	6	**0.339**
berlin52	7	4019	8	19.39	3999	10	**0.382**
	9	3430	7	26.08	3514	7	**0.502**
	11	3742	5	0.22	3261	6	**0.207**
st70	7	297	10	232.24	289	10	**0.372**
	9	271	9	173.87	262	10	**0.374**
	11	269	7	13.21	261	8	**0.331**
eil76	7	241	11	1.15	226	13	**1.502**
	9	193	9	7.43	204	10	**0.584**
	11	180	8	30.48	185	8	**0.342**
pr76	7	53255	11	54.2	50896	13	**4.243**
	9	45792	10	6743.66	45672	11	**3.656**
	11	45955	7	0.11	43573	10	**3.373**
rat99	7	572	14	22.74	534	17	**1.025**
	9	462	12	1749.66	457	13	**0.531**
	11	456	10	88.87	454	12	0.899
kroA100	7	10306	15	6303.03	10203	16	2.067
	9	9573	12	524.49	9512	13	**1.4**
	11	9460	10	409.45	9316	11	1.681
kroB100	7	11123	14	45.62	10473	16	**1.945**
	9	9505	12	2112.57	9948	13	**2.302**
	11	9049	10	1056.27	9196	10	3.727
kroC100	7	10367	15	3391.82	10451	17	**1.443**
	9	9952	12	35.91	9592	14	**1.871**
	11	9150	10	1389.84	8866	11	1.92
kroD100	7	11085	14	10.29	10599	15	**1.972**
	9	10564	11	6.18	9316	14	1.931
	11	9175	10	968.39	9055	13	1.230
kroE100	7	11323	15	1971.32	10542	17	**2.351**
	9	9095	12	1918.72	9161	13	**1.585**
	11	8936	10	609.81	8761	11	1.206
rd100	7	4105	14	24.43	3844	16	0.773
	9	3414	12	1798.14	3386	13	**0.831**
	11	3453	10	8.6	3050	12	0.717
kroA150	7	12367	22	2252.5	12543	26	**3.412**
	9	11955	17	2454.99	11193	20	**3.856**
	11	10564	15	5483.07	10300	17	8.215
kroB150	7	12876	21	196.85	12782	25	**3.875**
	9	11774	18	2760.03	11278	21	**5.725**
	11	10968	14	26.64	10246	17	2.187

Name	*Current and Schilling [4]*				*Proposed MGA_GPX*		
	NC	*Best obj.*	*NB*	*TB*	*Best obj.*	*NB*	*TB*
kroA200	7	14667	28	537.6	15014	33	**8.639**
	9	12683	23	1504.07	13215	25	**6.527**
	11	12736	19	398.25	**11955**	22	**7.108**
kroB200	7	14952	29	365.08	**14865**	33	**4.23**
	9	13679	23	637.66	**12987**	26	**3.496**
	11	12265	20	493.64	12342	22	**4.466**
No. best		11			37		
Average		9805.917		1017.941	**9548.1**		**2.302**

Table 5

Comparison of results of CSP using proposed MGA GPX with MA [5]

	Avg. length	*Avg. Time*
MA	9025.54	1.59
MGA_GPX	10304.21	**0.40**
Error in MGA_GPX	Increased by 14%	Decreased by 25%

5. CONCLUSION

The goal of the proposed algorithm for CSP is to cover all the customers in a devastated area by traversing a subset of facilities where a node may be a customer or a facility which can be applied in real life scenarios such as after natural as well as artificial disasters. The results show the superiority of proposed meta-heuristic algorithm compared to two other existing algorithms. In future, this problem can be extended for uncertain environment, where the cost of the edges or the demand of the tour will not be represented by crisp values.

6. REFERENCES

1. R. A. Jensen, Mass fatality and casualty incidents: a field guide. CRC Press, 1999.
2. G.B. Dantzig, R. Fulkerson, S.M. Johnson, "Solution of a large scale traveling salesman problem", Oper. Res. vol. 2, no. 4, pp. 393–410, 1954.
3. G. Gutin and A. P. Punnen, The traveling salesman problem and its variations. Springer Science & Business Media, vol. 12, 2006.
4. J. R. Current and D. A. Schilling, "The covering salesman problem," Transportation science, vol. 23, no. 3, pp. 208–213, 1989.
5. M. H. Shaelaie, M. Salari, and Z. Naji-Azimi, "The generalized covering traveling salesman problem," Applied Soft Computing, vol. 24, pp. 867–878, 2014.
6. E. M. Arkin and R. Hassin, "Approximation algorithms for the geometric covering salesman problem," Discrete Applied Mathematics, vol. 55, no. 3, pp. 197–218, 1994.
7. R. Patterson and E. Rolland, "The cardinality constrained covering traveling salesman problem," Computers & Operations Research, vol. 30, no. 1, pp. 97–116, 2003.
8. F. Zhao, S. Li, J. Sun, and D. Mei, "Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem," Computers & Industrial Engineering, vol. 56, no. 4, pp. 1642–1648, 2009.
9. S. Ghafurian and N. Javadian, "An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems," Applied Soft Computing, vol. 11, no. 1, pp. 1256–1262, 2011.
10. X. Wen, Y. Xu, and H. Zhang, "Online traveling salesman problem with deadline and advanced information," Computers & Industrial Engineering, vol. 63, no. 4, pp. 1048–1053, 2012.

11. X. Bao and Z. Liu, "An improved approximation algorithm for the clustered traveling salesman problem," Information Processing Letters, vol. 112, no. 23, pp. 908–910, 2012.

12. D. Karapetyan and G. Gutin, "Efficient local search algorithms for known and new neighborhoods for the generalized traveling salesman problem," European Journal of Operational Research, vol. 219, no. 2, pp. 234–251, 2012.

13. G. Laporte, A. Asef-Vaziri, and C. Sriskandarajah, "Some applications of the generalized travelling salesman problem," Journal of the Operational Research Society, vol. 47, no. 12, pp. 1461–1467, 1996.

14. M. Gendreau, G. Laporte, and F. Semet, "The covering tour problem," Operations Research, vol. 45, no. 4, pp. 568–576, 1997.

15. M. Hachicha, M. J. Hodgson, G. Laporte, and F. Semet, "Heuristics for the multi-vehicle covering tour problem," Computers & Operations Research, vol. 27, no. 1, pp. 29–42, 2000.

16. B. Golden, Z. Naji-Azimi, S. Raghavan, M. Salari, and P. Toth, "The generalized covering salesman problem," INFORMS Journal on Computing, vol. 24, no. 4, pp. 534–553, 2012.

17. M. Salari, M. Reihaneh, and M. S. Sabbagh, "Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem," Computers & Industrial Engineering, vol. 83, pp. 244–251, 2015.

18. M. Dorigo, "Optimization, learning and natural algorithms," Ph. D. Thesis, Politecnico di Milano, Italy, 1992.

19. G. Ozbaygin, H. Yaman, and O. E. Karasan, "Time constrained maximal covering salesman problem with weighted demands and partial coverage," Comput. Oper. Res., vol. 76, no. C, pp. 226–237, Dec. 2016.

20. G. Reinelt, "Tsplib—a traveling salesman problem library," ORSA journal on computing, vol. 3, no. 4, pp. 376–384, 1991.

21. M. L. Ryerkerk, R. C. Averill, K. Deb, and E. D. Goodman, "Solving metameric variable-length optimization problems using genetic algorithms," Genetic Programming and Evolvable Machines, pp. 1-31, 2016.