Research Article

Covering Solid Travelling Salesman Problem - An Algorithamic Study

Anupam Mukherjee*, Samir Maity**
Goutam Panigrahi*, Manoranjan Maiti******

Abstract

In this paper we model a Covering Solid Travelling Salesman Problem (CSTSP), a generalization of a two dimensional Covering Sales-man Problem (CSP) which is reduced to a Solid Travelling sales man problem (STSP) and solved by using a modified Genetic Algorithm (MGA). A salesman starts from an initial node and visits a subset of nodes only for once so that unvisited nodes are within a predetermined distance from the visited nodes, and comes back to the initial node. Here, a set of conveyances is available at the above mentioned nodes and the salesman uses the appropriate conveyance for minimum cost. Thus the problem reduces to finding the optimal covering set of nodes and the proper conveyance for travel so that total travel cost is minimum. This reduced STSP is solved by an MGA, which consists of roulette-wheels election, cyclic crossover, 2-inversemutation. A teach selected node, a random mutation for vehicle is introduced. Hence the CSTSP is solved by a random search (RS) for covering set and MGA, i.e, RS-MGA. The model is illustrated with some randomly generated cost and distance matrices.

Keywords: Solid TSP, Covering Salesman Problem, Modified GA.

Introduction

Travelling Salesman Problem (TSP) is one of the most well-known NP-hard problems, which can be defined as follows: Let N be a set of cities, c_{ij} be the cost to node j from node i. The objective is to minimize the total tour cost when a salesman starts from an initial node, visits all the nodes exactly once and comes back to the initial node. In 1930, this problem was formulated. Different types of

*Department of Mathematics, National Institute of Technology Durgapur, Durgapur, West Bengal - 713209, India, mukherjee.anupam.bnk@gmail.com
**Department of Computer Science, Vidyasagar University, West Bengal - 721102, India, maitysamir13@gmail.com
***Department of Mathematics, National Institute of Technology Durgapur, Durgapur, West Bengal - 713209, India, panigrahi_goutam@rediffmail.com
****Former Professor, Department of Applied Mathematics, Vidyasagar University, West Bengal - 721102, India., mmaiti2005@yahoo.co.in

TSPs were developed and generalized by many researchers, e.g, TSP with precedence constraints (Moon, 2002), stochastic TSP (Chang, 2009), symmetric TSP (Mestria, 2019), asymmetric TSP (Majumder, 2011), etc.

Covering Salesman Problem (CSP) is a generalization of TSP which was first introduced by Current and Schilling (Current, 1989). In CSP, unlike TSP, salesmen need not visit all the given nodes but a subset of nodes such that all other nodes be covered within a predetermined distance from the visited nodes. They developed a simple heuristic method to solve this problem consisting of two parts, first is the unicost set covering problem (SCP) to find a minimum number of nodes which can cover all unvisited nodes, and next to run TSP over different SCP solutions (if exist) to obtain a tour having a minimum cost. Two local search (LS) algorithms LS1 and LS2 (Golden, 2012) and an integer programming based LS (Salari, 2012) was proposed for CSP and later, a combined algorithm including ant colony optimization (ACO) and dynamic programming technique for the same problem was developed by Salari et al. (Salari, 2012).

Genetic Algorithm (GA) is an optimization procedure inspired by evolution theory. Different types of GA have been developed by many researchers, e.g., Adaptive GA (Xudong, 2013), Hybrid GA (Zhao, 2009), NSGA II, etc (DebP, 1995).

Solid traveling salesman problem (STSP) is same as TSP, in which there are several types of conveyances at each node which salesman (SM) can avail. Considering different types of conveyances and obstacles in each city, Changdar et al. (Changdar, 2013) introduced the notion of STSP in crisp and fuzzy environment. Later, Maity et al. (Maity, 2015) generalized the problem further in bi-random and random-fuzzy environments. They developed genetic algorithms to solve their models.

Covering Solid Traveling Salesman Problem (CSTSP) can be defined as CSP, in which there are several types of conveyances at each node for travel. The variety of conveyances may be different at different nodes. Till now, none has investigated this type of realistic problem.

Given a node set N. A salesman starts his tour from any one node and visits a subset N' of N, each node exactly once, such that all unvisited nodes are covered within a predetermined length from the nodes on tour and returns to the starting node, and at each node, the salesman chooses any one of the available vehicles bearing different travel cost. Considering multiple vehicles, we solve the above mentioned CSTSP in two steps. For the unicost set cover problem, we propose a simple heuristic method (random search (RS) algorithm) which inserts nodes randomly (each node can be included only once) and checks feasibility of set cover, i.e., whether the unvisited nodes are covered within a covering distance from the visited nodes or not. This yields naturally many solutions having a different number of nodes in a defined time interval. We mark the solutions with a minimum number of the node. With these set of nodes, the problem is reduced to find the solution of STSP, which is solved by a modified genetic algorithm (MGA).

In MGA, roulette-wheel selection, cyclic crossover, 2-inverse mutation and random mutation for vehicles (at each node) are introduced. Thus the CSTSP is solved by a combined RS-MGA technique. Randomly generated 100 100 and 100 100 3 matrices for distances and costs respectively are used for illustration of the model.

The proposed MGA is tested with some standard test problems and results of Salari et al. 's (Salari, 2012) CSP are compared by proposed RS-MGA.

Mathematical Formulations

Covering Salesman Problem

Given a complete graph G=(N,A), minimize the total tour cost such that a salesman starts from an initial node, visits a subset N' ⊂ N of nodes in which no node is visited more than once and comes back to the initial node, so that each of the nodes not on tour are in a predetermined distance from at least one of the visited nodes. The mathematical formulation of this problem may be stated as:

$$Minimize \;\; Z = \sum_{i=1}^{|N|} \sum_{j=1}^{|N|} c_{ij} x_{ij} \tag{1}$$

Subject to:

$$\sum_{i=1}^{|N|} \sum_{j \in D_l} x_{ij} \geq 1, \forall \; l \in N \tag{2}$$

$$\sum_{i=1}^{|N|} x_{ik} = \sum_{j=1}^{|N|} x_{kj} = 0 \;\; or \;\; 1, \forall \; k \in N \tag{3}$$

$$x_{ij} \in \{0, 1\} \tag{4}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \;\; \forall \; S \subset N' \subset N, \;\; 2 \leq |S| \leq |N'| - 2 \tag{5}$$

where, N' is the set of visiting nodes, c_{ij} is the cost from the node i to the node j,

$$x_{ij} = \begin{cases} 1, \text{ if } \exists \text{ a path from the node } i \text{ to the node } j, \\ 0, \text{ otherwise;} \end{cases}$$

$D_l = \{j \colon d_{lj} \leq \Delta_j\}$, d_{ij} = shortest distance between i and j, Δ_j = maximum covering distance at node j.

Equation (1) minimizes the total tour cost. Equation (2) ensures every node of the graph is either visited or is covered by the visited nodes. Equation (3) indicates that for a vertex, in degree and out degree are the same. Equation (4) implies the binary character of x_{ij} and (5) is the sub tour elimination constraint, i.e., if a node is on tour, it is visited not more than once (only the initial node is visited twice).

The above equation (1) – (5) can be rewritten as follows:
Let $N = \{x_1, x_2, x_3, \ldots, x_{|N|}\}$ be the set of nodes. Determine a complete tour $(x_{\alpha_1}, x_{\alpha_2}, x_{\alpha_3}, \ldots, x_{\alpha_m}, x_{\alpha_1})$, $m \le |N|$ to

$$\text{minimize} \sum_{i=1}^{m-1} c(x_{\alpha_i} x_{\alpha_{i+1}}) + c(x_{\alpha_m} x_{\alpha_1}); \tag{6}$$

$$\text{such that, } x_j \in \bar{B}(x_{\alpha_i}, \Delta_{\alpha_i}), \;\; \forall \;\; x_j \in N \text{ and for some } i; \tag{7}$$

where $\alpha_i \in \{1, 2, 3, \ldots, |N|\}$ and $\alpha_i \ne \alpha_j$ for $i \ne j$, $c(i, j) = c_{ij}$, $\bar{B}(a, r)$ means closed disc with center a and radius r, Δ_j = maximum covering distance at node j.

Covering Solid Traveling Salesman Problem
The problem may be stated as follows: Given a complete graph G=(N,A), minimize the total tour cost when a salesman starts from an initial node and visits a subset N'⊂N of nodes exactly once using any one of the available vehicles at each node and returns to the initial node such that all unvisited nodes are covered within a predetermined distance from the visited nodes. Let at each node p number of conveyances be available and c(i,j,k) be the cost from the node i to node j using the kth vehicle, (k {1,2,3,…,p}). Then the mathematical formulation is given below:

Let $N = \{x_1, x_2, x_3, \ldots, x_{|N|}\}$ be the set of nodes and $V = \{v_1, v_2, v_3, \ldots, v_p\}$ be the set of vehicles. Determine a complete tour $(x_{\alpha_1}, x_{\alpha_2}, x_{\alpha_3}, \ldots, x_{\alpha_m}, x_{\alpha_1})$, $m \le |N|$ to

$$\text{minimize} \sum_{i=1}^{m-1} c(x_{\alpha_i} x_{\alpha_{i+1}}, v'_{\alpha_i}) + c(x_{\alpha_m} x_{\alpha_1}, v'_{\alpha_m}); \tag{8}$$

such that,

$$x_j \in \bar{B}(x_{\alpha_i}, \Delta_{\alpha_i}), \;\; \forall \;\; x_j \in N \text{ and for some } i. \tag{9}$$

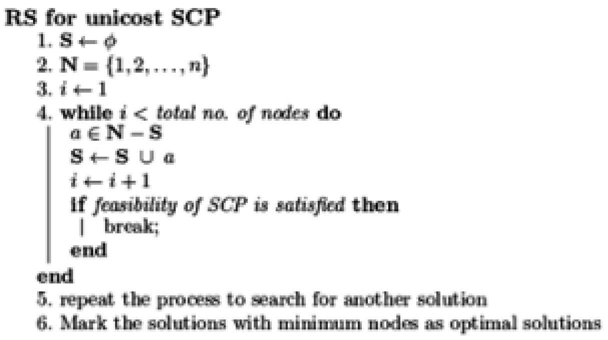
where $\alpha_i \in \{1, 2, 3, \ldots, |N|\}$ and $\alpha_i \ne \alpha_j$ for $i \ne j$, $v'_{\alpha_i} \in V$, $\forall \;\; \alpha_i \in \{1, 2, 3, \ldots, |N|\}$, $c(i, j, k) = c_{ijk}$, $\bar{B}(a, r)$ means closed disc with center a and radius r, Δ_j = maximum covering distance at node j.

Solution Procedure

Proposed Algorithm: The proposed CSTSP is solved by RS-MGA method, consisting of the two steps: at first we solve the unicost set cover problem (SCP) to obtain minimum number of nodes that covers all unvisited nodes and in the next step, the solutions obtained from SCP are used for TSP (using MGA) to get the minimum cost tour.

RS for Set cover

In this procedure, we start by considering an arbitrary node in a null set and check the feasibility for set cover in each step. We keep adding a new node until the feasibility condition is satisfied. If at a certain step, condition for set cover is satisfied, we take that set cover result and search, within a predefined time interval, for another solution. If more than one solution is found in the time interval, we mark the solutions with a minimum number of nodes as optimal SCP solutions. We propose the algorithm as follows:



```
RS for unicost SCP
  1. $S \leftarrow \phi$
  2. $N = \{1, 2, \ldots, n\}$
  3. $i \leftarrow 1$
  4. while $i <$ total no. of nodes do
       $a \in N - S$
       $S \leftarrow S \cup a$
       $i \leftarrow i + 1$
       if feasibility of SCP is satisfied then
         break;
       end
  end
  5. repeat the process to search for another solution
  6. Mark the solutions with minimum nodes as optimal solutions
```

Algorithm 1: Algorithm of RS for unicost SCP

Modified Genetic Algorithm for STSP

We modify a GA for STSP which includes roulette-wheel selection, cyclic crossover, 2-inverse mutation and random mutation (at each node) for the conveyances.

Initialization

In GA for solving TSP, a chromosome is represented by a complete tour. Let m be the number of chromosomes and n be the number of nodes, p be the number of vehicles and the vehicle set be $V=\{v_1,v_2,\ldots,v_p\}$. Then each chromosome X_i, (i=1,2,…,m) represented as $X_i=(x_i1,x_i2,\ldots,x_in)$ and the corresponding vehicle set is represented as $V'_i=(v'_i1,v'_i2,\ldots,v'_in)$.

For initialization, we make such chromosomes by generating random numbers from 1 to n such that each node occurs exactly once in a chromosome. The algorithm for initialization is as follows:

```
Data: Number of chromosomes m, number of nodes n
Result: A set of m chromosomes each having n different nodes
while $i$=1 to $m$ do
    while $j$=1 to $n$ do
        label: $t = rand[1, n]$
        for(k=1 to j-1)
        if $t$=$x_{ik}$ then
            goto label
        else
            $x_{ij} = t$
        end
        $temp = rand[1, p]$, $v'_{ij} = v_{temp}$
    end
end
```

Algorithm 2: Algorithm for initialization

The algorithms of conventional roulette-wheel selection, cyclic crossover, proposed 2-inverse mutation and random mutation for vehicles are given below:

Roulette-wheel Selection:

```
Data: pop-size ($m$), population set
Result: mating pool
1. calculate the fitness, say $f_i$ of each chromosome
2. calculate $\sum_{i=1}^{m} f_i$
3. calculate probability of selection $p_i$ of each chromosome
4. calculate cumulative probability of selection $q_i$ of each chromosome
5. for $i$=1 to $m$ do
    r=rand[0,1]
    if $q_k < r \leq q_{k+1}$ then
        select $k + 1^{th}$ chromosome in the new $i^t h$ position
    end
end
6. replace the old chromosomes with new ones
```

Algorithm 3: Algorithm for roulette-wheel selection

Cyclic Crossover

```
Data: number of nodes $n$, parent1, parent2
Result: offspring1, offspring2
1. r=rand[1,n]
2. offspring1[r]=parent2[r]
3. for i=1 to r-1 and i=r+1 to n do
    if offspring1[r]=parent1[i] then
        offspring1[i]=parent2[i]
        r=i
        goto step 3
    else
        stop
    end
end
4. s=rand[1,n]
5 offspring2[s]=parent1[s]
6.for i=1 to s-1 and i=s+1 to n do
    if offspring2[r]=parent2[i] then
        offspring2[i]=parent1[i]
        s=i
        goto step 5
    else
        stop
    end
end
```

Algorithm 4: Algorithm for cyclic crossover

2-inverse Mutation

```
Data: number of nodes n, chromosome
Result: mutated chromosome
1. generate r1=rand[1,n] and r2=rand[1,n] such that $r1 < r2$
2. for i=r1 to r1+r2/2 do
    $node[i] \leftarrow node[r2 - i + r1]$
end
3.Repeat the steps 1 and 2.
```

Algorithm 5: Algorithm for 2-inverse mutation

```
Data: number of nodes $n$, chromosome, maximum number of vehicles $p$, cost
      matrix
Result: chromosome with mutated vehicles
for $i = 1$ to $n$ do
    $temp = rand[1, p]$,
    if $cost(x_i, x_{i+1}, v_{temp}) \leq cost(x_i, x_{i+1}, v_i')$ then
        replace $v_i'$ by $v_{temp}$
    end
end
```

Algorithm 6: Algorithm for random mutation for vehicles

Procedure MGA for STSP

Thus the algorithm of MGA used for the solution of STSP is as follows:

```
Data: Maximum number of generation (maxgen), pop-size, number of nodes,
      cost matrix, $p_c$, $p_m$
Result: Minimum tour cost
1. Initialization of chromosomes
2. Set $gen \leftarrow 1$, $glob\text{-}best = loc\text{-}best = MAX\text{-}INT$
3. Selection procedure
4. for $i$=1 to pop-size do
    if $rand[0,1] < p_c$ then
        $i^{th}$ is selected for crossover
    end
end
5. procedure crossover among the mating pools
6. for $i = 1$ to pop-size do
    if $rand[0,1] < p_m$ then
        select $i^{th}$ chromosome for mutation and mutation for vehicles
    end
end
7. Procedure mutation and random mutation for vehicles


8. for $i = 1$ to pop-size do
    if $cost[i] < loc\text{-}best$ then
        $loc\text{-}best = cost[i]$
    end
end
9. $gen \leftarrow gen + 1$
10. if $loc\text{-}best < glob\text{-}best$ then
    $glob\text{-}best \leftarrow loc\text{-}best$
end
11. if $gen < maxgen$ then
    goto step 3
else
    goto step 12
end
12. end
```

Algorithm 7: Algorithm of MGA for STSP

RS-MGA algorithm for CSTSP

Ultimately, the algorithm of RS-MGA used for the solution of the proposed CSTSP is as follows:

```
Data: number of nodes $n$, distance matrix $[d_{ij}]_n$, cost matrix $[c_{ijk}]_{n\times n\times p}$,
      covering distance matrix $[\Delta_{ij}]_n$
Result: complete tour with minimum cost such that visited nodes cover all
        unvisited nodes
1. solve the unicost SCP for $[d_{ij}]_n$ using $[\Delta_{ij}]_n$
2. for $i$= 1 to total no. of SCP solutions do
    $mincost \leftarrow STSP[SCP[i]]$
    if $mincost > STSP[SCP[i+1]]$ then
        $mincost \leftarrow STSP[SCP[i+1]]$
    end
end
```

Algorithm 8: Algorithm of RS-MGA for CSTSP

Numerical Experiments

Verification with earlier results

As we discussed earlier, the proposed problem (CSTSP) contains two parts, unicost set cover and STSP. To justify the efficiency of the MGA algorithm implemented in C language, we picked up some benchmark problems from TSBLIB and compared with actual results of those in Table 1. Some test TSP problems are solved using RS-MGA, reducing the nodes to covering sets. Salari et al. [SalariR] used some test TSP problems and solved using a hybrid algorithm consisting of Dynamic Programming and Ant Colony Optimization (ACO) (i.e., reducing the nodes to covering sets). Results of some of these problems are obtained by RS-MGA algorithm and are compared in Table 2.

Table 1:
Algorithm tested with benchmark problems [TSPLIB]:

Problem	Best known result	MGA Result	Generation	time(s)
gr17	2085	2085	169	0.62
fri26	937	937	464	0.78
bayg29	1610	1610	531	0.82
bays29	2020	2020	695	0.94
dantzig42	699	721	917	1.07
eil51	426	436	1135	1.21

Table 2:
Comparison with Salari et al.'s [SalariR] results:

Problem	No. of nearest nodes	Salari's method	MGA Result	time(s)(60s +)
eil51	7	164	**158**	1.14
	9	159	**157**	1.03
	11	**147**	149	1.02
berlin52	7	**3887**	3891	1.10
	9	3430	**3362**	1.14
	11	3262	**2832**	1.26
st70	7	**288**	292	1.76
	9	259	**241**	2.05
	11	247	**233**	1.84
eil76	7	207	**184**	2.74
	9	186	**173**	1.27
	11	170	**161**	2.42
pr76	7	**50275**	51277	2.41
	9	45348	**42916**	3.38
	11	43028	**42607**	2.72
rat99	7	486	**453**	4.67
	9	455	**441**	3.92
	11	444	**423**	3.46
kroA100	7	**9674**	10558	4.54
	9	9159	**8860**	4.11
	11	**8901**	9316	4.36

Proposed experiment

For computational results of the proposed CSTSP, we generated a 100×100 distance matrix with lower bound 1 and upper bound 20 and a 100×100×3 cost matrix with lower bound 1.75 and upper bound 37. The cost matrix implies 100 nodes, each having ≤ 3 vehicles with different costs. In the first step, the unicost SCP was solved using the distance matrix and 4 distance units as maximum covering distance at each node with time-bound 60 seconds. In the next step, solid TSP was solved for all the above SCP solutions obtained in that time bound using MGA. The path with minimum cost among all solutions of SCP is considered as the near optimal solution of the CSTSP problem, some of which (10 best solutions) are given in Table 3.

Table 3:
Near optimal solutions of the proposed model:

Optimized covering path (nodes/vehicles)	Cost	time(s)(60s+)
15/2 27/2 55/0 64/0 94/2 2/1 21/1 48/1 5/2 59/1	37.50	3.62
72/2 69/0 74/2 30/0 22/2 5/1 84/0 76/0 26/0 98/1	49.55	–
60/1 79/2 10/2 94/2 9/2 72/2 69/0 18/1 34/1 59/2	57.15	–
96/2 9/2 57/0 15/0 25/1 89/1 73/0 35/1 13/1 95/1	58.90	–
21/1 86/0 5/2 37/0 20/1 85/2 81/1 27/2 38/1 53/0	61.45	–
94/0 9/2 0/1 58/1 59/1 21/1 48/2 89/0 11/1 53/2	61.90	–
54/2 90/0 4/0 35/2 59/0 16/0 56/2 94/0 7/1 26/0	65.55	–
66/1 18/2 27/2 96/2 34/0 94/2 29/0 33/2 85/0 80/2	65.60	–
7/0 57/2 56/1 70/0 24/1 59/0 30/2 37/0 27/0 18/2	69.95	–
98/1 72/0 43/2 68/0 27/0 46/0 38/1 21/2 5/0 92/2	70.10	–

Discussion

From Table 1, it is observed that the proposed MGA algorithm gives the same results for the first four TSP problems. For the last two problems whose order sizes are large, the proposed algorithm failed to give exact results, furnishes slightly higher results than the standard results.

In Table 2, results by the RS-MGA have compared with the Salari et al.'s (SalariR, 2012) hybrid algorithm for some TSP problems, which are reduced to CSP by their hybrid ACO algorithm. In most of the cases, our proposed algorithm furnishes better results than them.

In Table 3, some near-optimal solutions of the proposed CSTSP for the given cost and distance matrices are presented along with the optimal paths. It is seen that for the given data, the minimum number of covering sets within 4 distance units is 10. Here 10 consecutive near-optimal solutions are given because due to some reasons, if the optimal path bearing lowest cost cannot be implemented, then the decision maker (DM) can go for the next or some other convenient path.

In this proposed RS-MGA, the RS gives minimum 10 nodes to cover the network and the optimal tour cost is found among those SCP solutions, though more than 10 nodes may give lesser cost, in that case, the number of covering nodes may be relaxed up to some more nodes.

Conclusion

In the present investigation, for the first time, an algorithm is presented for the covering solid traveling salesman problem. This is very much useful for the big merchant houses, government officials and other several real-world problems. Suppose there are several sales offices of a big business house in a state. A high sales official wants to give direction/to explain the company's policy to the salesman at each sales office. Instead of visiting each office, the high sales official decides to visit some specified offices and asks the others to come to one of these specified offices as per their distances. Obviously, there may be several types of conveyances at the specified nodes and the sales official is supposed to choose the appropriate path and vehicles for the minimum cost.

Also, this model may be used for rural health care delivery problems. Practically it is not possible to deliver health care services to all the cities or villages in a part of a country affected by any of natural disasters or some other causes. The service may be delivered to some of the cities using any of the available vehicles at each city so that people from the other areas can come to their nearest cities for their needs.

The proposed problem can be extended to include a restriction on some particular nodes for inclusion (due to some factors). The proposed algorithm also can be tested with other types of selection such as probabilistic selection, etc., crossover such as order crossover, comparison crossover, etc., and mutations.

References

- Chang, T., Wan, Y., & Tooi, W., *A stochastic dynamic travelling salesman problem with hard time windows. European Journal of Operational Research, 198(3), 748â“-759 (2009).*
- Changdar, C., Maiti, M. K., & Maiti, M., *A Constrained solid TSP in fuzzy environment:two heuristic approaches. Iranian Journal of Fuzzy System, 10(1), 1-â“28 (2013).*
- Current, J. R., & Schilling, D. A., *The covering salesman problem. Transportation Science, 23(3), 208â“-213 (1989).*
- Deb, & Agarwal, R. B., *Simulated Binary Crossover for Continuous Search Space. Complex Systems, 9, 115-â“148 (1995).*

- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T., *A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197, (2002).*
- Golden, B. L., Naji-Azimi, Z., Raghavan, S., Salari, M., & Toth, P., *The generalized covering salesman problem. INFORMS Journal on Computing, 24(4), 534â"553 (2012).*
- Hachicha, M., Hodgson, M. J., Laporte, G., & Semet, F., *Heuristics for the multi-vehicle covering tour problem. Computers & Operations Research, 27, 29-â"42 (2000).*
- Khanra, A., Maiti, M. K., & Maiti, M., *Profit maximization of TSP through a hybrid algorithm. Computers & Industrial Engineering, 88, 229â"-236 (2015).*
- Maity, S., Roy, A., & Maiti, M., *A Modified Genetic Algorithm for solving uncertain Constrained Solid Travelling Salesman Problems. Computers & Industrial Engineering 83, 273–296 (2015).*
- Majumder, A. K., & Bhunia, A. K., *Genetic algorithm for asymmetric traveling salesman problem with imprecise travel times. Journal of Computational and Applied Mathematics, 235(9), 3063â"-3078 (2011).*
- Mestria, M., Ochi, L. S., & Martins, S. L, *GRASP with path relinking for the symmetric Euclidean clustered traveling salesman problem. Computers and Operations Research, 40(12), 3218â"-3229 (2013).*
- Moon, C., Ki, J., Choi, G., & Seo, Y., *An efficient genetic algorithm for the traveling salesman problem with precedence constraints. European Journal of Operational Research, 140, 606â"-617 (2002).*
- Salari, M., & Naji-Azimi, Z., *An integer programming-based local search for the covering salesman problem. Computers & Operations Research, 39, 2594â"2602 (2012).*
- Salari, M., Reihaneh, M., & Sabbagh, M. S., *Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem. Computers & Industrial Engineering 83, 244â"-251 (2015).*
- TSPLIB, *http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/.*
- Xudong, S., & Yunlong, X., *An Improved Adaptive Genetic Algorithm International Conference on Education Technology and Management Science (ICETMS) (2013).*
- Zhao, F., Sun, J., Li, S., & Liu., W., *A Hybrid Genetic Algorithm for the Traveling Salesman Problem with Pickup and Delivery. International Journal of Automation and Computing, 06(1), 97–102 (2009).*