ISSN (e): 2250 – 3005 || Volume, 09 || Issue, 6 || June– 2019 ||
International Journal of Computational Engineering Research (IJCER)

An efficient Hybrid Algorithm for the Multiple Traveling Salesmen Problem using novel Crossover and Ant Colony Optimization

Dharm Raj Singh[1], Manoj Kumar Singh[2]
[1]Assistant Professor,[1]Department of Computer Applications Jagatpur P. G. College Varanasi-221302
[2]Associate Professor Department of Computer Science Banaras Hindu University, Varanasi-221005, India,
Corresponding Author: Dharm Raj Singh

ABSTRACT
This paper, we proposed a new hybrid algorithm in the framework of genetic algorithm (GA) and ant colony optimization (ACO). In this technique which uses Ant colony optimization for tour construction and novel crossover and 2-opt mutation with multi chromosome representation using for solving the multiple traveling salesmen problem (MTSP) for near-optimal solutions. The proposed crossover and 2-opt mutation is used to refine the solution obtained from ACO. Multi chromosome representation technique is a proven to minimize the problem search space. Moreover this representation is more similar to characteristic of MTSP. We evaluate and compare the proposed hybrid technique with four different crossover methods for two MTSP objective functions, namely, minimizing total travel distance and minimizing longest tour. The performance of proposed hybrid algorithm is better in terms of best solution quality and average solution. The experimental result shows that the proposed hybrid algorithm can improve the solution quality of the GA. Experimental result shows that the proposed algorithm is more efficient, than algorithm developed by other.

KEYWORDS: *Multiple Traveling Salesman problems, Ant Colony Optimization, Proposed Crossover operator.*

Date of Submission: 05-06-2019 Date of acceptance:20-06-2019

I. INTRODUCTION

1.1 Multiple Traveling Salesman problem

The multiple traveling salesmen problem (MTSP) is generalization of the traveling salesman problem (TSP), In TSP a salesman visits all of the cities exactly once and back to the starting city whereas in MTSP, The MTSP can be defined as follows: Given a set of nodes (cities), let there be m salesmen located at a single source node (city). The remaining nodes (cities) that are to be visited are called intermediate nodes. Then, the MTSP consists of finding tours for all m salesmen, who all start and end at the source node (city), such that each intermediate node is visited exactly once by only one salesman and back to the source node (starting city) and the total cost of visiting all nodes is minimized [30].

Given a collection of n cities and the distance (travel cost/distance) of travel between every pair of city must be partitioned into m tours for m>1 salesmen to serve a set of n>m cites. The objective of MTSP is to find the set of closed tour with minimum cost of visiting all of the cities exactly once and back to the starting city. The objective is minimizing the total cost (distance) traveled by all salesman and minimize the maximum distance traveled by any one salesman. This problem can be modeled using graph theory, where cities are represented by node and roads between the two cities are represented by edges and the cost of travel from one city to another city will be the weight of edges. This type of problem is modeled in graph theory. Given a weighted graph G = (V, E, w), where V is the set of node representing cities, E is the set of edges representing roads and w is weight (cost/distance) between each pair of nodes. A closed tour in which all the vertices are distinct which is known as Hamilton cycle. Finding set of m Hamiltonian cycles with minimum travel cost in the weighted graph gives the desired solution. i. e. total cost of visiting all nodes is minimized. The cost between cities is represented by matrix, known as cost matrix $C = (c_{ij})$, i, j = 1, 2, · · ·, n. In cost matrix C the $(i,j)^{th}$ entry c_{ij}, represents the cost

of travel from i[th] to j[th] city. The matrix C is said to be symmetric when $c_{ij}=c_{ji}$, ∀ (i, j) ∈ E and asymmetric otherwise. Based on Integer Linear Programming, formulation for the MTSP is defined as follows:
Let the decision variables

$$x_{ij} = \begin{cases} 1, & \text{if salesman travels } i \text{ from to } j \text{ city} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The objective function is

$$\min imize \quad Z = \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij} + mc_m \quad (2)$$

Subject to

$$\sum_{i=2}^{n} x_{i1} = m \quad (3)$$

$$\sum_{j=2}^{n} x_{1j} = m \quad (4)$$

Since, each city can be visited only once, we have

$$\sum_{i=1}^{n} x_{ij} = 1,\ j = 1, 2, \ldots, n;\ i \neq j \quad (5)$$

Again, since the salesman has to leave each city except city n, we have

$$\sum_{j=1}^{n} x_{ij} = 1, i = 1, 2, \ldots, n; i \neq j \quad (6)$$

$$+ subtour\ elimination\ constraints, \quad (7)$$

Where $x_{ij} \in \{0,1\}$ is a binary variable whose value indicates whether a salesman visits next city or not. If variable $x_{ij} = 1,$ a salesmen selects an edge on the tour for travel, if variable $x_{ij} = 0,$ a salesmen does not selects an edge on the tour for travel, Equation (2) describes the total cost to be minimized and C_m represents the cost of one salesman's participation. Equation (3) and (4) ensures that exactly m salesmen depart from and returns back to node 1. Equation (5) describes that a salesman can only enter the city j for one time, Equation (6) describes that a salesman can only departure from the city i for one time, i.e. Equation (5) and (6) describes that a salesman visits each city only once. Equation (7) is used to prevent sub-tours, which are degenerate tours that are formed between intermediate nodes and not connected to the origin.

1.2 Literature Review

Since TSP and MTSP belongs to a class of NP-hard combinatorial optimization problems, which has many applications in different engineering and optimization problems. In recent years, many heuristic or meta-heuristic algorithms have been developed for solving the NP-hard optimization problems, such as Simulated Annealing [8], Genetic Algorithms [10], [11] and Particle Swarm Optimization [9], etc. The most common application of MTSP are in the area of scheduling and routing e.g. military search and rescue operations [1], interview scheduling [2], A columnar competitive model (CCM) of neural networks incorporates with a winner-take-all learning rule was employed to solve the MTSP [3], workload balancing [5], mission planning [31], School bus routing problem [7], Crew scheduling [32], design of global navigation satellite system (GNSS) surveying networks[6] etc.
Finding global solution for TSP problem within affordable computational resources is difficult. Instead of global optimal solution, a suboptimal solution with reasonable computational load is obtained and it will further pruned for global solution using heuristic concept. A comprehensive survey on heuristic methods is available in [12], [13] and references there in GA is an iterative, population based, heuristic search technique for solving MTSP. Tang et al. [14] proposed a GA with one chromosome representation for MTSP problem to solve the hot rolling production scheduling problem. Song et al. [15] proposed an extended simulated annealing approach for the MTSP. Malmborg [16] and Park [17] solved MTSP using genetic algorithm has focus on vehicle scheduling problem. The Carter and ragsdale [18] modeled as the MTSP for vehicle scheduling problem that uses a new two part chromosome representation. Brown et al. [19] proposed a grouping GA that uses a chromosome presentation. Singh and Baghel [20] proposed a grouping GA that uses different chromosome representation and different crossover and mutation. Liu et al. [21] presented an ant colony optimization algorithm for MTSP. Yousefikhoshbakht [24] proposed modification of the ant colony optimization for solving MTSP. Yuan et al. [22] proposed a new crossover approach for MTSP with two part chromosome crossover (TCX) using GA.

Venkatesh and Singh [23] proposed two meta-heuristic approaches based on artificial bee colony and invasive weed optimization algorithm for the MTSP

.

II. CHROMOSOME REPRESENTATIONS FOR THE MTSP

In this section, we introduced different techniques for chromosome representation, which are commonly used when solving MTSP using GAs. We also discussed their properties, weaknesses and benefits for solving MTSP.

2.1 One Chromosome Technique

In the one chromosome representation a solution for the MTSP represented by using a single chromosome of length n+m −1, where n represents a permutation (tour) of n cities with integer value ranging from 1 to n and m represents number of salesman. The solution chromosome is divided in to m sub tours by inserting m −1 dummy negative integers that represents the change from one salesman to another. Therefore number of possible solutions are (n + m-1)!. However, many of the possible chromosomes are redundant. The one-chromosome technique is given in [4, 19]. For example in Fig.1, (Where n = 10 and m = 3) the first salesman will visit cities 1, 3 and 10, the second salesman will visit cities 9, 5 and 8, and the third salesman will visit cities 7, 4, 6 and 2.

Cites

1	3	10	-1	9	5	8	-2	7	4	6	2

Salesman 1 Salesman 2 Salesman 3

Fig.1. Representation of one chromosome technique for a 10 city MTSP with three salesmen

4.1 Two Chromosome Technique

In the two chromosome representation a solution for the MTSP represented by using a double chromosome of length n. one-chromosome represents a permutation (tour) of n cities with integer value ranging from 1 to n and the other chromosome represents city assigned to a salesman. The two-chromosome technique is given in [18], [19], [20]. For example in Fig.2, (Where n = 10 and m = 3) the first salesman will visit cities 1, 2, 6 and 9, the second salesman will visit cities 5, 7 and 10, the third salesman will visit cities 3, 4 and 8. In this method there are $n!m^n$ possible solutions to the problem where n is the number of cities and m is the number of salesmen. However, many of the possible solutions are redundant. For example, given in Fig.2 the first two genes in each of the chromosomes can be interchanged to create different chromosomes that result in an identical (or redundant) solution.

Cities

1	3	5	9	7	4	8	6	10	2

Salesmen

1	3	2	1	2	3	3	1	2	1

Fig.2. Representation of the two chromosome technique for a 10 city MTSP with three salesmen.

4.1 Two-part chromosome technique

In the two-part chromosome representation a solution for the MTSP represented by using a two part of the chromosome. The first part of the chromosome of length n represents a permutation (tour) of n cities which value is integer from 1 to n and second part of chromosome of length m gives the number of cities assigned to each salesman. Therefore, in this representation the total length of the chromosome is n + m. The values assigned to the second part of the chromosome are constrained to be m positive integers that sum are equal to number of cities to be visited. The two part chromosome technique is given in [18], [19]. For example in Fig.3, the first salesman will visit cities 1, 3 and 5, the second salesman will visit cities 9, 7, 4, and 8, the third salesman will visit cities 6, 10 and 2. The two-part chromosome technique reduces redundant solutions. Therefore the size of the search space is also reduced. In the two-part chromosome technique for the MTSP, there are $n!$ possible permutations for the first part of the chromosome. The second part of the chromosome represents a positive vector of integers $(k_1, k_2, \ldots, k_m)$ that must sum equal to n. There are $\binom{n-1}{m-1}$ distinct positive integer- valued m-vectors that satisfy this requirement [22]. Thus, size of the solution space for the two-part chromosome representation is $n!\binom{n-1}{m-1}$.

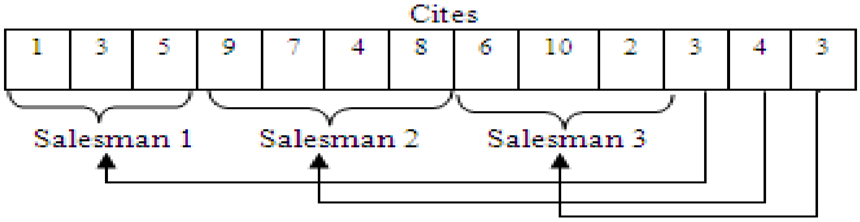

Fig.3. Representation of two parts chromosome technique for a 10 city MTSP with three salesmen.

4.1 Multi-chromosome technique

In this method MTSP solution represents by a multi-chromosome. The length of first chromosome is k_1, the length of second chromosome is k_2, and so on. Therefore total number of cities in multi-chromosome representation equals to $\sum_{i=1}^{m} k_i = n$. The multi-chromosome technique is given in [28]. For example in Fig.4, (Where n = 10 and m = 3) the first salesman will visit cities 2, 5 and 7, the second salesman will visit cities 10, 1, 8 and 9, the third salesman will visit cities 6, 4 and 3. Determining number of cites of the first chromosome is equal to the problem of obtaining an ordered subset of k_1 elements from a set of n elements. There is $\frac{n!}{(n-k_1)!}$ distinct assignment. For the second chromosome the assignment is $\frac{(n-k_1)!}{(n-k_1-k_2)!}$, and so on. Therefore total search space of the multi-chromosome technique can be formulated as follows:

$$\frac{n!}{\cancel{(n-k_1)!}} * \frac{\cancel{(n-k_1)!}}{\cancel{(n-k_1-k_2)!}} * \ldots * \frac{\cancel{(n-k_1-\ldots-k_{m-1})!}}{(n-k_1-\ldots-k_m)!} = \frac{n!}{(n-n)!} = n! \tag{8}$$

The length of the each chromosome represents a positive vector of integers $(k_1, k_2, \ldots, k_m)$ that must sum to n. There are $\binom{n-1}{m-1}$ distinct positive integer- valued m-vectors that satisfy this requirement [22]. Thus, size of the solution space for the multi-chromosome representation is $n!\binom{n-1}{m-1}$. It is equal to the two part chromosome representation but this approach is more similar to characteristics of MTSP. So it can be more problem specific therefore more effective. So our proposed method used multi-chromosome technique that gives better performance.

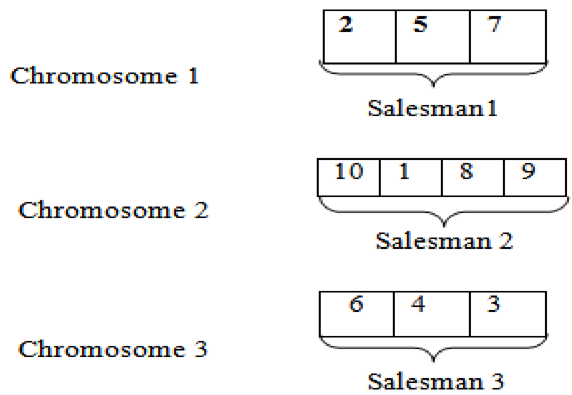

Fig.4: multi-chromosome technique for a 10 city MTSP with three salesmen.

III. PROPOSED METHOD

The proposed hybrid method is an approximate method that depends on good heuristics alternate adequately between exploration and exploitation. The overall procedures of our hybrid algorithm that combines the Ant colony optimization tour construction, proposed crossover and 2-opt mutation heuristics. In the proposed algorithm, the main idea of algorithm is to create an initial population of tours by using simple Ant colony optimization tour construction heuristic. On use of ACO, for initializing population, the exploration space for solution is reduced and hence the search time, in contrast to random generation of population of tours/approximate solution. In ACO, the initial population of size P = m by using m ants , m ants start from m distinguish city to travel all city and return to its starting city in such manner that, each city travel only once except the starting city. The first step of algorithm is to generate an initial population of path by using Ant colony optimization tour construction heuristic. Also randomly generate population of break-points of tours for assigning number of nodes to each salesman for each tour. In the second step of algorithm fitness value is obtained for each tour in population. In the third step, randomly select five tours and its corresponding break-points from generated population and population of break-points. After then replace first tour and its corresponding break-points with minimum cost among selected five tours and its break- points for crossover. In the fourth step, apply proposed crossover operator on the first two tours from the selected five tours with crossover probability rate (pc) after then generate two new children tour and its break-points. In the last step, apply 2-opt optimal mutation operator with minimum cost between new individuals that generated after crossover. After then update the population and its corresponding break-points. These processes can be repeated until termination condition is satisfied.

3.1 Ant Colony Optimization

Ant Colony Optimization is a population-based general search technique for the solution of difficult combinatorial problems, which is inspired by the pheromone trail laying behavior of real ant colonies. An important behavior of the ants is the foraging behavior and how they are able to find the shortest path between sources and their nest despite being almost blind. When ants move, ants can leave a chemical which called pheromone trail on the path to transmit information. The amount of pheromone corresponds to the quality of the solution found by the ants; and visibility information represents some forms of heuristic information, which is combined with the pheromone value in order to make decisions. Ants tend to choose the paths marked by the strongest pheromone concentration. The indirect communication of the ants is to pheromone trails in order to enable them to find shortest path between source and destination. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph.

Main characteristic of Ant System is that, at each iteration, the pheromone values are updated by all the m ants that have built a solution in the iteration itself c_{ij} (i, j = 1, 2,…, n) is the travel cost(distance) between cities. The amount of pheromone residual between i^{th} city and j^{th} city is denoted as $\tau(i, j)$ .The heuristic function from i^{th} city to j^{th} city is denoted as η (i, j), which is equal to $1/c_{ij}$. Initially, ants were assigned to cities randomly. Setting each path's initial pheromone to be equal, $\tau(i, j)$ = constant. Ants move from i^{th} city to j^{th} city or select next j^{th} city with probability P_{ij}^{k} is as follow:

$$P_{ij}^{k} = \begin{cases} \dfrac{[\tau(i,j)]^{\alpha} * [\eta(i,j)]^{\beta}}{\sum [\tau(i,j)]^{\alpha} * [\eta(i,j)]^{\beta}}, if \quad j \in J_k, \\ 0, \ else \end{cases} \tag{9}$$

where α and β are the information stimulating factor and expect factor, respectively. The method of updating pheromone is given in [21], [33]. The pheromone τ_{ij} associated with the edge joining from i^{th} city to j^{th} city, is updated as follows:

$$\tau_{ij} = (1-e)\tau_{ij} + \sum_{k=1}^{m} \Delta \tau_{ij}^{k}, \tag{10}$$

where $\Delta \tau_{ij}$ is equal to Q/L_k , if city (i, j) belong to tour otherwise zero and L_k is the length of the tour constructed by ant k. 'e' is the evaporation rate; m is the number of ants and Q is a constant.

3.2 Proposed crossover operator

In proposed crossover, the first city of chromosome s_1 is copied to the first position on child c_1 and the first city of chromosome s_2 is copied to the first position on child c_2. The remaining cites changed as procedure given in Fig. 6. For example in Fig. 5, the first position of parents s_1, s_2 are copied to the first position in child c_1 and c_2 respectively, remaining positions 2, 3, 4, 5, 6, 7, 8 and 9 cities are swapped as procedure given in Fig. 6.

Parent s_1 :	2	1	6	5	4	3	8	7	9
Parent s_2 :	4	3	2	8	7	9	5	6	1
Child c_1 :	2								
Child c_2 :	4								
Child c_1 :	2	1	3	8	7	6	5	4	9
Child c_2 :	4	3	2	8	7	9	5	6	1

Fig.5: proposed crossover

```
c1=zeros(1,n);
c2=zeros(1,n);
c1(1)=s1(1);
c2(1)=s2(1);
m=1;
for i=1:n;
    if(s1(1)~=s2(i));
        for j=2:n;
         if(s2(m)==s1(j));
            c1(j)=s1(j);
           m=j;
         end
       end
    end
end
 for i=1:n
    if ( c1(i)==0)
         c1(i)=s2(i);
    end
end
m=1;
for i=1:n
    if(s2(1)~=s1(i))
        for j=2:n
         if(s1(m)==s2(j))
            c2(j)=s2(j);
           m=j;
         end
       end
    end
end
for i=1:n
    if ( c2(i)==0)
         c2(i)=s1(i);
    end
end
```

Fig.6: Algorithm for proposed crossover

3.3 2-Opt optimal mutation

Basically, 2-opt mutation replaces two edge from a tour by two new edges that are not in tour such that the cost of new tour is less than that of the original tour [25][26]. This replacement process is continued till no further improvement in the cost of the new tour is possible, this is often referred to as 2-opt optimal. Note that 2-opt mutation keeps the feasible tour corresponding to a reversal of a subsequence of the cities.

The method proceeds by replacing two non-adjacent edges (v_i, v_{i+1}) and (v_j, v_{j+1}) by two new edges (v_i, v_j) and (v_{i+1}, v_{j+1}), which are the only other two edges that can create a tour when the first two are deleted. In order to maintain a consistent orientation of the path by the predecessor-successor relationship, one of the two sub paths remaining after dropping the first two edges must be reversed [27]. For example, inverting the sub tour $(v_{i+1}, ..., v_j)$ in tour $(v_i, v_{i+1}, ..., v_j, v_{j+1})$ using 2-opt mutation, we get the tour $(v_i, v_j, ..., v_{i+1}, v_{j+1})$. Finally, the change in the cost of the tour $(v_i, v_{i+1}, ..., v_j, v_{j+1})$ and $(v_i, v_j, ..., v_{i+1}, v_{j+1})$ is given by:

$$\Delta_{ij} = c(v_i, v_j) + c(v_{i+1}, v_{j+1}) - c(v_i, v_{i+1}) - c(v_j, v_{j+1}). \qquad (11)$$

The change in cost, Δ, gives the clue of the improvement of the tour. This process is repeated till Δ is negative. 2-opt process can be generalized to perform k-opt mutation that removes some k edges and adds k new edges. There are $\binom{n}{k}$ possible ways to remove k edges in a path and $[(k-1)!] * 2^{k-1}$ ways to reconnect the disconnected sub-paths (including the initial path) to recover the tour structure [27].

In the proposed algorithm, the initial suboptimal solution is generated with the help of Ant colony optimization tour construction heuristic. The ACO tour construction heuristic reduces redundant solutions than random generation tour. And also, we use 2-opt mutation, which is sequential move and has less diversify population for next generation. But by using the combination of ACO tour construction heuristic and 2-opt optimal mutation, the exploration space for optimal tour gets reduced and hence the search time as well. Therefore the combination of ACO tour construction heuristic and 2-opt mutation refines the tour for global optimality and decreases time to get the optimal solution. We found that the proposed method ACO+2opt mutation gives better result in comparison to other heuristic methods reported in [4], [22].

Table 1 Computational test conditions

Number of cites(n)	Number of salesmen(m)	GA generations
51	3, 5 and10	5000
100	3, 5, 10 and 20	10000
150	3, 5, 10, 20 and 30	20000

Table 2 Experimental results for minimising total travel distance

problem	crossover	M=3			M=5			M=10			M=20			M=30		
		Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best
MTSP-51	Proposed	**441**	**4**	**434**	**469**	**5**	**461**	**604**	**10**	**580**						
	Group+GA	466	6	460	515	10	499	693	20	669						
	TCX	510	24	466	536	26	499	636	17	602	-	-	-	-	-	-
	ORX+A	584	29	517	621	39	551	709	33	648	-	-	-	-	-	-
	CYX+A	591	43	511	622	44	530	710	42	633	-	-	-	-	-	-
	PMX+A	601	38	513	606	40	537	705	34	625	-	-	-	-	-	-
MTSP-100	Proposed	**21807**	**176**	**21565**	**24489**	**662**	**23275**	**31957**	**1092**	**29249**	**54111**	**1566**	**50831**			
	Group+GA	24071	690	22959	26220	755	24559	35943	1221	33136	67623	2038	62963	-	-	-
	TCX	32708	2267	28943	34179	2006	30941	36921	1964	32802	46976	1773	44112	-	-	-
	ORX+A	41516	3356	36713	42716	2806	36196	44631	2997	38717	54265	3059	47971	-	-	-
	CYX+A	41911	3195	35791	43634	2804	35421	45150	3241	40894	52916	2884	46466	-	-	-
	PMX+A	41441	3423	33802	42063	3931	33908	44786	3467	39785	52142	2588	46212	-	-	-
MTSP-150	Proposed	**40641**	**873**	**37957**	**40753**	**810**	**38714**	**47751**	**1307**	**42234**	**68334**	**975**	**66424**	**90254**	**1413**	**87136**
	Group+GA	40697	826	39504	42639	1825	39862	55895	1765	50892	79734	1138	77668	105072	922	102880
	TCX	55851	2588	51126	61596	4759	51627	61360	3888	54473	69701	4340	62456	84008	5285	76481
	ORX+A	67037	3745	60090	68018	3377	62539	72113	3637	63899	81696	5372	71933	96122	4562	88515
	CYX+A	67463	4454	55335	69860	4342	31521	71584	4845	63126	83471	4197	75146	97106	3911	89008
	PMX+A	68152	5140	58303	69112	4011	60761	72620	4334	64975	81178	4920	73281	95752	4923	87402

Table 3 Experimental results for minimising longest tour**.**

problem	crossover	M=3			M=5			M=10			M=20			M=30		
		Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best	Mean	SD	Best
MTSP-51	Proposed	**177**	**5**	**160**	**134**	**4**	**118**	**112**	**0**	**112**						
	Group+GA	188	4	184	139	3	129	112	0	112	-	-	-	-	-	-
	TCX	207	13	182	153	10	135	113	2	112	-	-	-	-	-	-
	ORX+A	216	12	191	165	11	139	128	13	112	-	-	-	-	-	-
	CYX+A	222	16	188	161	12	138	131	16	112	-	-	-	-	-	-
	PMX+A	218	11	191	161	10	141	130	12	112	-	-	-	-	-	-
MTSP-100	Proposed	**10055**	337	**8509**	8045	219	**7607**	**6644**	83	**6358**	**6358**	0	**6358**			
	Group+GA	10384	194	10031	**7907**	137	7728	6688	65	6581	6404	43	6363	-	-	-
	TCX	14365	1013	12645	10086	674	8730	7768	492	6796	6768	433	6358	-	-	-
	ORX+A	15137	1462	12997	11459	1053	9415	9286	1385	7373	8123	881	6666	-	-	-
	CYX+A	15759	1242	13467	11333	1278	9507	9151	1364	7111	8109	936	6516	-	-	-
	PMX+A	15238	1371	12249	11233	1177	9267	6890	1482	7187	8265	8608	6570	-	-	-
MTSP-150	Proposed	**15008**	257	**14380**	**10596**	**288**	10149	7374	104	7019	5930	79	5740	5410	63	**5246**
	Group+GA	15389	334	14804	13077	2539	**10106**	**6884**	**99**	**6684**	**5546**	36	**5483**	**5251**	5	5248
	TCX	22523	1226	20556	16054	1227	14096	10722	927	8475	9640	789	8423	8759	806	7169
	ORX+A	24766	1689	22015	17646	1519	15266	15150	2006	11788	13669	1816	10274	12204	1376	9182
	CYX+A	23906	1941	20915	17608	1563	14029	14738	1750	10779	14111	1872	8365	13091	1295	10694
	PMX+A	24216	1802	20347	17741	1235	15418	14489	2139	10738	13810	1315	11722	12608	1667	10080

IV. EXPERIMENTAL RESULTS

4.1 Experimental setup

For evaluating the performance of experimental results, Intel(R) Core (i5) 3.20 GHz processor, 4GB RAM on MATLAB is used. In this experiment, test problems are selected benchmark instances taken from the TSPLIB. The test problems are Euclidean, two-dimensional symmetric problem with 51, 100, and 150 cities. These problems are denoted as MTSP-51, MTSP-100, and MTSP150 cities. Table 1 represents the experimental conditions of 12 different problem sizes (n) and salesmen (m) combinations along with the run time for each type of problem. The termination criterion is the number of generations. The values of parameters used in experiment are: population size and number of ants(m) =100, $\alpha = 1$, $\beta = 1.5$, initial evaporation (e) = 0.01, tournament selection =5, probability of mutation (pm) = 0.2 and crossover probability rate =0.85. The performance of methods compared based on best, mean and standard-deviation (SD).

4.2 Experimental results and analysis

In this paper, we compared the proposed algorithm ACO+2opt mutation with recent algorithm as presented in [4], [22]. To evaluate the benefits of the proposed method, computational experiments were conducted to compare the performance of four crossover methods on a set of problems created for the MTSP. We compare the proposed method ACO+2opt mutation with other crossover methods like Group+GA, TCX, ORX +A, CYX +A and PMX +A. We conducted experiments for two objective functions. The first objective function is to minimize the total travel distance of all the salesmen. The second objective function for MTSP is minimizing the longest individual tour, which is also called make-span [28], [29]. Minimizing the longest tour balances the cities (or workload) among the salesmen and also minimizes the distance travelled by the salesmen. Minimization of the longest tour affects the fitness value as it gets decreased with the increasing number of salesmen. The test problems are Euclidean, two-dimensional symmetric problems with 51, 100, and 150 cities. In this paper we refer to these test problems as MTSP-51, MTSP- 100, and MTSP-150, respectively. We perform 30 trails for our method. The comparative results are shown in Table 2 and Table 3. The best performance of the method for particular instance has shown in bold. It can be seen from the Table 2 compare the average solution quality of proposed method, Group+GA, TCX, ORX+A, CYX+A, PMX+A on Instances of MTSP51, MTSP100, MTSP150, whereas from Table 3 compare the minimising of maximum distance solution quality of proposed method, Group+GA, TCX, ORX+A, CYX+A, PMX+A on Instances of MTSP51, MTSP100, MTSP150. From the Table 2, it is clear that our proposed method gives better performance in terms of total travel distance for all instances with respect to all performance parameter. From the Table 3, it is clear that our proposed method gives better performance in terms of minimising longest tour for all instances with respect to all performance parameters, except MTSP150 for Best and Mean for m=10, MTSP150 for Best and Mean for m=20 and MTSP150 for Best and Mean for m=30.

V. CONCLUSION

In this paper, we have discussed a new proposed hybrid algorithm ACO+2opt mutation for the MTSP. It is the combination of Ant colony optimization for tour construction heuristic, proposed crossover and 2-opt mutation. We overcome the random initialization contamination by using ACO tour construction method. However, crossover and mutation used to maintain local optimality. Our proposed hybrid method has also a proposed crossover with a powerful local improvement 2-opt mutation heuristic. This combination affects the quality of the solution. The performance comparison is shown in Table 2 and Table 3. By comparing the results of the proposed hybrid algorithm with the standard algorithms as presented in [4], [22], it can be concluded that the proposed hybrid algorithm gives better performance. Furthermore, proposed algorithm can be applied to other versions of the MTSP and combinatorial optimization problems like vehicle routing problem, school bus routing problem and the sequencing of jobs for better result.

REFERENCES

[1]. Sheldon H. Jacobson, Laura A. McLay, Shane N. Hall, Darrall Henderson, Diane E. Vaughan, "Optimal search strategies using simultaneous generalized hill climbing algorithms", Mathematical and Computer Modelling, 43, 2006, pp. 1061–1073.

[2]. Gilbert, Kenneth C., and Ruth B. Hofstra. "A new multiperiod multiple traveling salesman problem with heuristic and application to a scheduling problem." Decision Sciences 23.1 (1992): 250-259.

[3]. H. Qu, Z. Yi, H. Tang, A columnar competitive model for solving multi-traveling salesman problem Chaos, Solitons and Fractals, 31, (2007) 1009-1019.

[4]. Singh, Dharm Raj, et al. "Genetic Algorithm for Solving Multiple Traveling Salesmen Problem using a New Crossover and Population Generation." Computación y Sistemas 22.2 (2018).

[5]. Okonjo-Adigwe, C. "An effective method of balancing the workload amongst salesmen." Omega 16.2 (1988): 159-163.

[6]. Saleh, Hussain Aziz, and RachidChelouah. "The design of the global navigation satellite system surveying networks using genetic algorithms."Engineering Applications of Artificial Intelligence 17.1 (2004): 111-122.

[7]. Angel RD, Caudle WL, Noonan R, Whinston A. Computer assisted school bus scheduling. Management Science 1972;18:279–88.

[8]. Chen, Yong, and Pan Zhang.: Optimized annealing of traveling salesman problem from the nth-nearest-neighbor distribution,

[9]. Marinakis, Yannis, and Magdalene Marinaki. "A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem." Computers & Operations Research 37.3 (2010): 432-442.

[10]. Albayrak .M, N. Allahverdi,: Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms, Expert Syst. Appl. 38 (2011) 1313–1320.

[11]. Louis, Sushil J., and Gong Li.: Case injected genetic algorithms for traveling salesman problems, Information Sciences 122.2 (2000): 201-225

[12]. Gutin, Gregory, and Abraham P. Punnen, eds. The traveling salesman problem and its variations. Vol. 12. Springer Science & Business Media, 2006.

[13]. Johnson, David S., and Lyle A. McGeoch. "The traveling salesman problem: A case study in local optimization." Local search in combinatorial optimization 1 (1997): 215-310.

[14]. Tang, Lixin, et al. "A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex." European Journal of Operational Research 124.2 (2000): 267-282.

[15]. Song, Chi-Hwa, Kyunghee Lee, and Won Don Lee. "Extended simulated annealing for augmented TSP and multi-salesmen TSP." Neural Networks, 2003. Proceedings of the International Joint Conference on. Vol. 3. IEEE, 2003.

[16]. Malmborg, Charles J. "A genetic algorithm for service level based vehicle scheduling." European Journal of Operational Research 93.1 (1996): 121-134.

[17]. Park, Yang-Byung. "A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines." International Journal of Production Economics 73.2 (2001): 175-188.

[18]. Carter, Arthur E., and Cliff T. Ragsdale. "A new approach to solving the multiple traveling salesperson problem using genetic algorithms." European journal of operational research 175.1 (2006): 246-257.

[19]. Brown, Evelyn C., Cliff T. Ragsdale, and Arthur E. Carter. "A grouping genetic algorithm for the multiple traveling salesperson problem."International Journal of Information Technology & Decision Making 6.02 (2007): 333-347.

[20]. Singh, Alok, and Anurag Singh Baghel. "A new grouping genetic algorithm approach to the multiple traveling salesperson problem." Soft Computing13.1 (2009): 95-101.

[21]. Liu, Weimin, et al. "An ant colony optimization algorithm for the multiple traveling salesmen problem." 2009 4th IEEE conference on industrial electronics and applications. IEEE, 2009.

[22]. Yuan, Shuai, et al. "A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms." European Journal of Operational Research 228.1 (2013): 72-82.

[23]. Venkatesh, Pandiri, and Alok Singh. "Two metaheuristic approaches for the multiple traveling salesperson problem." Applied Soft Computing 26 (2015): 74-89.

[24]. Yousefikhoshbakht, Majid, Farzad Didehvar, and Farhad Rahmati. "Modification of the ant colony optimization for solving the multiple traveling salesman problem." Romanian Journal of Information Science and Technology 16.1 (2013): 65-80.

[25]. Gutin, Gregory, and Abraham P. Punnen, eds. The traveling salesman problem and its variations. Vol. 12. Springer, 2002.

[26]. Croes GA (1958) A method for solving traveling-salesman problems. Oper Res 6(6):791–812

[27]. Helsgaun, Keld, "An effective implementation of the Lin–Kernighan traveling salesman heuristic" European Journal of Operational Research 126 (2000) 106–130.

[28]. Andras Kiraly and janos Abonyi "optimization of MTSP by novel Representation based on genetic algorithm(2011)

[29]. Yuan, S., Skinner, B.T., Huang, S.D., Liu, D.K., Dissanayake, G., Lau, H., A job grouping approach for planning container transfers at automated seaport container terminals. Advanced Engineering Informatics 25, 413–426. Pagac, D., 2011.

[30]. Svestka JA, Huckfeldt VE. Computational experience with an m-salesman traveling salesman algorithm. Management Science 1973;19(7):790–9.
[31]. Brummit B, Stentz A. Dynamic mission planning for multiple mobile robots. Proceedings of the IEEE international conference on robotics and automation, April 1996.
[32]. Angel, R. D., et al. "Computer-assisted school bus scheduling." Management Science 18.6 (1972): B-279.
[33]. Marco Dorigo, Luca Maria Gambardella, "Ant colonies for the travelling salesman problem", BioSystems, 43, 1997, pp. 73–81.

Dharm Raj Singh" An efficient Hybrid Algorithm for the Multiple Traveling Salesmen Problem using novel Crossover and Ant Colony Optimization" International Journal of Computational Engineering Research (IJCER), vol. 09, no. 6, 2019, pp 62-71