Decision Science Letters 10 (2021) 525–534

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

An efficient genetic algorithm for solving open multiple travelling salesman problem with load balancing constraint

Purusotham Singamsetty[a] and Jayanth Kumar Thenepalle[b*]

[a]*Department of Mathematics, School of Advanced Sciences, VIT, Vellore-632014, Tamil Nadu, India*
[b]*Department of Science and Humanities, Sreenivasa Institute of Technology and Management Studies, Chittoor-517127, Andhra Pradesh, India*

C H R O N I C L E

Article history:
Received November 11, 2020
Received in revised format:
May 23, 2021
Accepted May 23 2021
Available online
May 27, 2021

Keywords:
Zero-one integer linear programming
Open multiple travelling salesman problem
Genetic algorithm
Load balancing constraint

A B S T R A C T

The multiple travelling salesman problem (MTSP) is one of the widely studied combinatorial optimization problems with various theoretical and practical applications. However, most of the studies intended to deal with classical MTSP, very limited attention has been given to an open multiple travelling salesman problem and its variants. In this paper, an open multiple travelling salesman problem with load balancing constraint (OMTSPLB) is addressed. The OMTSPLB differs from the conventional MTSP, in which all the salesmen start from the central depot and need not come back to it after visiting the given number of cities by accomplishing the load balance constraint, which helps in fairly distributing the task among all salesmen. The problem aims to minimize the overall traversal distance/cost for operating open tours subject to the load balancing constraint. A zero-one integer linear programming (0-1 ILP) model and an efficient metaheuristic genetic algorithm (GA), is established for the OMTSPLB. Since no existing study on OMTSPLB, the proposed GA is tested on the relaxed version of the present model, comparative results are reported. The comparative results show that the proposed GA is competent over the existing algorithms. Furthermore, extensive experiments are carried out on OMTSPLB and the results show that proposed GA can find the global solution effectively.

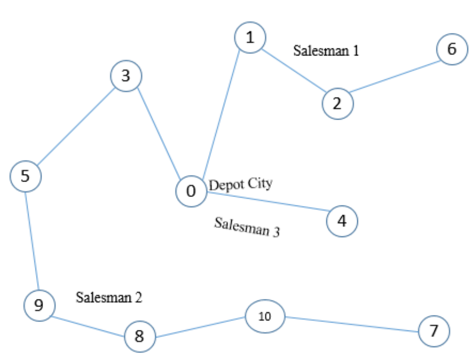
1. Introduction

Travelling salesman problem (TSP) is a typical NP problem arise in combinatorial optimization (Garey and Johnson, 1979), whose objective is to find the optimal salesman's tour such that the salesman visits all given cities only once, and return to the starting city at the end. Several practical problems can be designed as TSP after transformation, such as network communication (Bharath-Kumar & Jaffe, 1983), logistics distribution (Liu & Zhang, 2014), route planning (Ghadiry et al., 2015). From a graph theory viewpoint, the principal task of TSP is to obtain a least Hamiltonian cycle. The multiple travelling salesman problem (MTSP) is a generalized version of classical TSP, in which m salesman instead of single salesman are involved to cover $n(>m)$ cities. The MTSP looks for the splitting of n cities into m salesman clusters, so that each cluster of cities is covered by precisely one salesman, each city is covered once and only once and the overall distance covered by m salesman is minimized. The MTSP is more difficult than the TSP as it needs to find the optimal allocation of the set of the cities to each salesman. Several practical applications are formulated as the MTSP, which emerge in the areas of industry, business and engineering. To mention, job scheduling problem (Carter and Ragsdale, 2002), vehicle scheduling problem (VSP) (Carter & Ragsdale, 2006), Printing Press problem (Gorenstein, 1970), Crew scheduling and School bus routing (Király & Abonyi, 2011), surveying networks through Global navigation satellite system (GNSS) (Saleh & Chelouah, 2004), Workload balance (Okonjo-Adigwe, 1988) etc. Due to its wide applicability, the MTSP has been extensively studied and addressed several variants. Some of the variants namely, MTSP with fixed charges (Hong and Padberg, 1977), time windows (Kim and Park, 2004), Pickup and delivery (Wang and Regan, 2002), truncated MTSP

* Corresponding author.
E-mail address: jayanth.maths@gmail.com (J. K. Thenepalle)

doi: 10.5267/j.dsl.2021.5.003

(Bhavani, and Murthy, 2006), Multi-depots (Benevent & Martinez, 2013), Open close (Thenepalle & Singamsetty, 2019). Due to the NP-hard nature of the MTSP, it is required to employ heuristic techniques to tackle problems of larger size. Some of the heuristic techniques for MTSP, which were developed by adopting biological features or natural phenomena including, Evolutionary approach (Sofge et al. 2002), Modified imperialist competitive algorithm (MICA) (Larki and Yousefikhoshbakht, 2014), Ant colony optimization (ACO) (Yousefikhoshbakht, 2013), Neural networks (Modares et al. 1999), Particle swarm optimization (PSO) (Yan et al., 2012), Genetic algorithm (GA) (Király & Abonyi, 2011) to name but a few. However, several heuristics or metaheuristics were developed; each one has their complications. For instance, ACO is prone to have slower convergence rate, thus computationally expensive, PSO is likely to be stuck in local optimization and GA tends to have premature convergence and highly depends on the initial population. Although GA has its drawbacks, it is proven very effective and extensively used for solving MTSP (Xu et al. 2018). To review its progress Bailey (1967), the first who discussed the concept of GA and then Holland, did a systematic study on the mechanism of the survival of the fittest in 1975. Since then, the literature witnessed significant advancements of GA for solving MTSP and its variants. To mention the earlier works, Tang et al. (2000) suggested one-chromosome representation for the MTSP and used it to tackle hot rolling production scheduling problems. Carter and Ragsdale (2006) developed a GA that employs a two-part chromosome representation and relevant genetic operators. Brown et al. (2007) developed a grouping genetic algorithm (GGA) with one-chromosome and two-chromosome representations to solve the MTSP. Singh and Baghel (2009) then proposed a refined version of GGA which involves a steady-state population replacement model. Király and Abonyi (2011) studied MTSP and proposed a novel chromosome representation based GA. Yuan et al. (2013) introduced a new two-part chromosome representation for the GA to get near-optimal solutions of MTSP. However, this technique is affected by the development of chromosome length and the end solution. Kaliaperumal et al. (2015) proposed the improved two-part chromosome representation based GA to tackle MTSP. However, this technique allocates a distinct quantity of cities for each salesman, and thus, this study did not effectively address MTSP with load balancing constraint. Alves and Lopes (2015) addressed the MTSP with workload balance, developed GA to find optimal traversal distance by reducing the deviation among the distances covered by each salesman. Xu et al. (2018) proposed the two-phase heuristic algorithm (TPHA) which combines K-means and modified GA for solving MTSP subject to the workload balance. Lo et al. (2018) studied MTSP, proposed the GA in which two new local operators Branch and Bound and cross elimination effectively combined to find high-quality solutions within a short time. Recently, Harrath et al. (2019) studied MTSP, developed a hybrid algorithm, which integrates ACO, 2-Opt based GA (AC2OptGA) and showed that this algorithm outperforms other state-of-art techniques. Shuai et al. (2019) addressed a bi-objective MTSP model and suggested an evolutionary algorithm NSGA-II, which effectively jumps from the local optimum. A comparative study of various crossover operators of GA for MTSP can be found in Al-Omeer and Ahmed (2019). All these studies have inspired us to develop new GA for the OMTSPLB, which can find the best solutions within a short time. Based on the type of tours, the classical MTSP can be categorized into closed MTSP or simply MTSP and open MTSP. The closed MTSP means that a set of salesman starts from the central depot/starting city and needs to come back to the central depot after visiting the given cities, whereas the open MTSP (OMTSP) finds a set of Hamiltonian paths for salesman such that the overall traversal distance/cost is minimum. For instance, Fig. 1 illustrates a scenario defined by 3 salesman and 11 cities including depot city. Here, node 0 is the central depot from which all the salesman has to start visiting the cities and need not return back to it after visiting them. As shown in Fig.1, salesman 1 visits three cities, salesman 2 covers five cities, whereas the salesman 3 visits only one city. Clearly, the salesman's workload is not fairly distributed. As discussed above, the prime goal of OMTSP is to minimize the overall distance covered by all salesman, which results in an unbalanced workload model. Fig. 2 exhibits an appropriate arbitrary solution for OMTSP with workload balance. The main idea of load balancing is to fairly distribute the quantity of cities among all salesman. More precisely, if a problem is defined with m salesman and n cities, then the sufficient number of cities given for each salesman does not exceed $\left\lceil \frac{n-1}{m} \right\rceil$ cities. This constraint effectively controls the load balance in terms of cities allocation to each salesman. The present study addresses a new MTSP variant called *open multiple travelling salesman problem with load balancing constraint* (OMTSPLB). Inspired by the studied works cited above, GA is developed to solve the OMTSPLB.



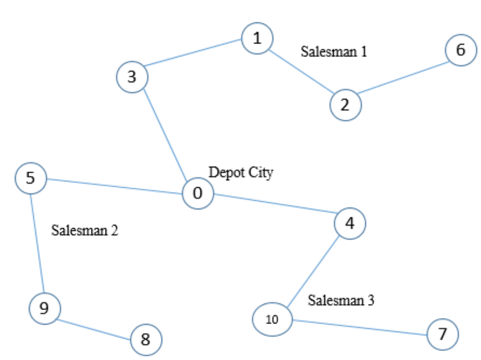


Fig 1. An example solution of OMTSP without load balancing

Fig 2. An example solution of OMTSP with load balancing

2. Problem Definition and Mathematical Formulation

The OMTSPLB can be described as follows: Let *G=G(E,V)* be the complete graph, where the vertex set $V=\{v_1,v_2,v_3,...,v_n\}$ denotes a set of n cities including one central depot /starting city (v_1) and the edge set $E=\{(v_i,v_j)/v_i,v_j\in V;\ i,j=1,2,...,n;\ i\neq j\}$ be the set of n^2-n edges. Each vertex v_i is specified with a position (x_i,y_i) in Cartesian coordinate system. Each edge (v_i,v_j) is given with a distance/cost $d_{ij}(d_{ij}=d_{ji};d_{ii}=\infty,d_{ij}>0)$, which is the Euclidean distance between the cities *i* and *j* . Let *K={1,...,m}* be the set with *m(m<n)* salesman positioned at the starting city (v_1). Each salesman starts from the starting city, takes a route and need not return to the starting city. Each city will be covered exactly once (except the starting city) by each salesman and each salesman can visit a maximum of Q $\left(Q=\left\lceil\frac{n-1}{m}\right\rceil\right)$ cities to achieve the load balance. The problem OMTSPLB aims to find m sequences of Hamiltonian paths over G with least distance/ cost. Here, the binary variable $x_{ij}^k\in\{0,1\}$, such that $x_{ij}^k=1$ if the k^{th} salesman visits j^{th} city from i^{th} city, and $x_{ij}^k=0$, otherwise. Here, an another binary variable $y_j^k\in\{0,1\}$ is introduced, such that $y_j^k=1$, if the k^{th} salesman visits j^{th} city and $y_j^k=0$, otherwise. Note that in the present study the starting city (v_1) is assumed as 1st city. The mathematical model for OMTSPLB is as follows:

$$\min Z=\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}\,x_{ij}^k \tag{1}$$

subject to

$$\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij}^k=n-1 \tag{2}$$

$$\sum_{k=1}^{m}\sum_{j=1}^{n}x_{1j}^k=m \tag{3}$$

$$\sum_{k=1}^{n}\sum_{i=1}^{n}x_{i1}^k=0 \tag{4}$$

$$\sum_{k=1}^{m}\sum_{i=1}^{n}x_{ij}^k=1;\ \ j\in V/\{1\} \tag{5}$$

$$\sum_{k=1}^{m}\sum_{j=1}^{n}x_{ij}^k\leq 1;\ \ i\in V/\{1\} \tag{6}$$

$$\sum_{i=1}^{n}y_i^k\leq Q;\ k=1,2,...,m \tag{7}$$

$$+\text{ subtour elimination constraints} \tag{8}$$

$$x_{ij}^k\in\{0,1\};\ i,j=1,2,...,n\ \ \&\ k=1,2,...,m \tag{9}$$

In the above model, the objective function (1) represents that the overall cost/distance of open paths with respect to salesman is minimum. Constraint (2) guarantees that any feasible solution should contain only number of edges. Constraint set (3-4) assures that all the salesman has to start from the starting city (here it is assumed as 1) and no salesman is required to return back to it, respectively. Constraint set (5-6) indicates that a salesman enters each city precisely once and departs from each city at most once, respectively. Constraint (7) represents that the total cities covered by any salesman does not exceed a specific value, which maintains load balance. The sub-tours, which are formed between intermediate cites and not included starting cities, are prevented by the constraint (8). Finally, Constraint (9) represents the binary variable $x_{ij}^k\in\{0,1\}$, such that $x_{ij}^k=1$ if the k^{th} salesman visits directly from i^{th} city to j^{th} city, and $x_{ij}^k=0$, otherwise.

3. Genetic Algorithm

In this section, first, the classical GA is described, and then the proposed algorithm is discussed in detail. The GA is one of the widely used metaheuristic algorithms in evolutionary computation research for solving combinatorial optimization problems. This algorithm is an adaptive searching technique based on the survival of the fittest strategy, was first discussed by Holland in 1975. In its nature, the GA starts with a set of initial solutions called the initial population, also referred to as chromosomes, in which all the genetic data is stored. Each numeral within the chromosome is treated as a gene. Further, a fitness value is determined to evaluate the performance of a chromosome. Each time, two chromosomes, called parent chromosomes are selected from the population randomly, which is proportionate to their fitness value. Then, the two chromosomes crossover to generate two new chromosomes for the subsequent generation. These new chromosomes will swap old ones if they have superior fitness values. Then, a mutation operation is applied to the newly produced chromosomes to maintain the diversity of the population. Repeat selection, crossover, and mutation processes to generate more chromosomes that are new until the newly generated population size equals to the old one. The iteration then starts with the new population. Since better chromosomes have a higher probability to be nominated for crossover and the new chromosomes generated to transmit the characteristics of their parent chromosomes. The search process continues for many generations until the predetermined criteria are met. For the OMTSPLB, the fitness value represents the overall traversal distance of all the salesmen. Hence, the lesser the fitness value results in the optimal/suboptimal solution.

3.1. Proposed GA

To find optimal/suboptimal solution to the OMTSPLB via GA, the key elements such as chromosome representation, population initialization, estimation of fitness value, selection, crossover, mutation operators and GA parameters are required. Different GA techniques might include distinct encoding, crossover and mutation operators, which results in divergence of the search process. Thus, it is essential to remodel the above operations to confirm that the optimal/suboptimal solution is indeed achieved. Below are the key elements in the proposed GA for OMTSPLB.

3.1.1. Chromosome Representation

An MTSP solution can be represented into a chromosome in several ways, including one chromosome (Tang et al., 2000), two chromosomes (Malmborg, 1996)), two-part chromosome (Carter & Ragsdale, 2006) etc. However, several studies include two-part chromosome strategy as it does the best in both solution quality and computational speed aspects compared to the former ones in solving MTSP and its variants. This inspires to adopt the two-part chromosome representation in this study. The name "two-part chromosome representation" is due to its structure, which has two parts. The initial part of the chromosome with length $n-1$ represents a permutation vector of integers from 2 to n. The first $n-1$ genes represent $n-1$ cities (without starting city) to be visited. The rest of the part with length m has m genes, and each gene is assigned with a value. The values assigned to these m genes are constrained to be positive integers, which lies between 1 and Q. Thus, the workload balance can be indeed achieved. Further, the sum of all the values in part II must be equal to $n-1$. Since all the salesmen must start from the central depot city $(1, \text{say})$, to save the memory usage it is excluded within the chromosome. A two-part chromosome solution for OMTSPLB with 11 cities and 3 salesmen is demonstrated in Fig.3. In this example, the salesman 1 visits 4 cities, i.e. $\text{Depot}(1) \to 4 \to 6 \to 7 \to 2$. Salesman 2 visits 3 cities i.e. $\text{Depot}(1) \to 11 \to 3 \to 5$ and finally, Salesman 3 visits 3 cities i.e. $\text{Depot}(1) \to 8 \to 10 \to 9$.

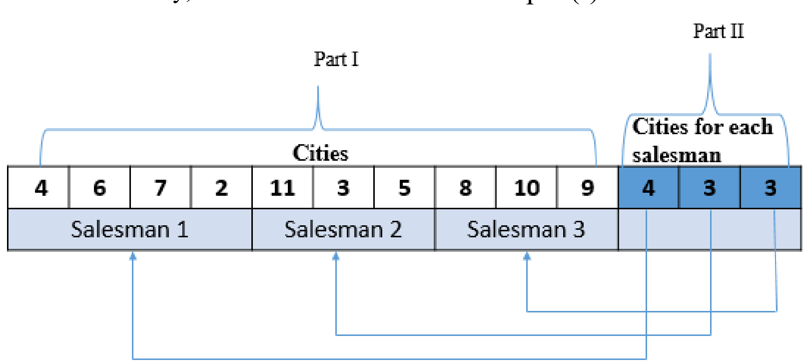


Fig 3. Representation of two-part chromosome for a 11 city OMTSPLB with 3 salesman

3.1.2. Initial Population Encoding

The efficiency of the GA certainly depends on the quality of the initial population. Therefore, it is inevitable to identify an appropriate encoding operator that decides the generation of the initial population. From the literature, it is evident that many GA techniques developed for MTSP and its variants, permutation encoding, are widely used to generate the initial population (Xu et al. (2018)). The chromosome representation in Fig. 3, specifies a single individual of the population called one of the solutions of the problem. The sequence of the integers are encoded into the genes, where each gene is associated with an integer also called a city. Thus, this kind of chromosome representation is so-called permutation encoding and the same is adopted in our study.

3.1.3. Fitness Function

The fitness function is used to compute individual chromosomes in the population. The selection strategy depends on the fitness value is an important step in GA. In particular, a chromosome with greater fitness value implies a better chance of being selected for the subsequent generation. In our study, the fitness function is assumed as the objective function specified by Eq. (1). Therefore, the chromosome corresponding solution with lesser distance/cost will possess a higher fitness value and thus have a greater genetic probability to be chosen.

3.1.4. Selection Operator

The selection operator is an essential step in the GA, as it affects its performance. In this article, the classical roulette wheel method is used as the selection operator of GA. This operator selects a chromosome from its population in a statistical fashion depending on its fitness value to enter into a reproducing pool. Those chromosomes closer to the solution have a better chance of being selected.

3.1.5. Crossover Operator

The crossover operator is again a key parameter, plays a significant role in GA efficiency and further helps to diversify the population. It is the process of mating/information sharing between two parent chromosomes. It combines the characteristics of two-parent chromosomes and produces two new chromosomes/offspring/child from them with the chance that good chromosomes may produce a superior child with best features. Several crossover operators have been developed for solving MTSP and its variants namely one and two-point crossover (Király and Abonyi (2011)), edge recombination crossover (Lo et al. (2018)), order crossover (Sedighpour et al. (2012)) etc. Recently, a multi-chromosome representation based modified distance preserving crossover operator is proposed (Singh et al. (2018)) for solving MTSP.

In this paper, we have adopted the strategy as discussed by Singh et al. (2018) and proposed a two-part chromosome based crossover operator, which is shown in Fig. 4. In this crossover, the first gene in part I of parent P1 is moved to the last gene in part I of child C2 and the last gene in part I of parent P1 is transferred to the first gene in part I of child C2. Similarly, the first gene in part I of parent P2 is copied into the last gene in part I of child C1 and the last gene in part I of parent P2 is copied in the first gene in part I of child C1. The rest of the genes transformed as shown in Fig. 4. Further, all the genes in part II of parents P1, P2 are copied as usual into genes in part II of child C2, C1, respectively. For instance, in Fig. 5, the first gene in part I of parent chromosomes P1, P2 (i.e. 4, 2) are copied into the last gene in part I of child C2, C1, respectively. Similarly, the last gene in part I of parent chromosomes P1, P2 (i.e. 9, 8) are copied into the first gene in part I of child C2, C1, respectively. Rest of the genes are exchanged as per the algorithm given in Fig. 4. Further, all the gene information (i.e. 4, 3, 3) in part II of parents P1, P2 are transformed into genes in part II of child C2, C1, respectively. The justification behind this crossover strategy depends on the idea that the city in optimal/suboptimal tours takes place in the same location. Thus, it helps with variability in the population.

```
C1     =zeros(1,n+m);
C2     =zeros(1,n+m);
C1(1)  =P2(1, n);
C2(1)  =P1(1, n);
C1(n)  =P2(1, 1);
C2(n)  =P1(1, 1);
       for i = 1: n
          for j = 2 : n-1
             if ( P2 (i)==P1(j))
                 C1(j) = P2(j);
             end
             if ( P1 (i)==P2(j))
                 C2(j) = P1(j);
             end
          end
       end
       for i=n+1:n+m
           C1(i)=P2(1, i);
           C2(i)=P1(1, i);
       end
```

Fig 4. Proposed crossover algorithm

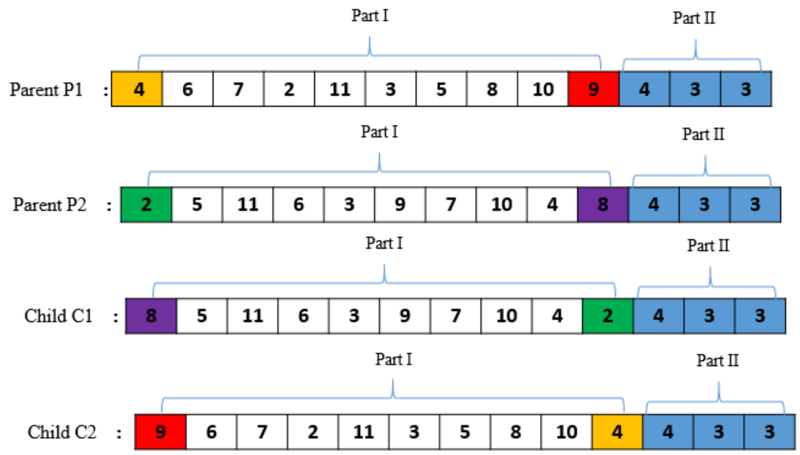


Fig 5. Proposed crossover operator

3.1.6. Mutation Operator

Mutation operation is executed next to the crossover. The intention of mutation is to avoid the GA from being trapped in a local optimum and enhance the genetic variability of the population. This work utilizes the complex mutation operator, which comprises Swap, Reverse swap/Flip and Slide mechanisms. All these mutation operators are employed to find the optimal distance and reduce computational time. With a mutation probability P_m, a parent chromosome is chosen. For a swapping operation, two different positions are selected randomly from the parent chromosome; the genes of these two positions are interchanged. For a reverse swap operation, two different positions are chosen to describe segment, the genes between these positions are reversed. Similarly, for a slide operation, two distinct positions are selected (say, i^{th} and j^{th} positions). Now, the new offspring can be produced by removing the gene in i^{th} position and copy the same in j^{th} position of the parent chromosome. Thus, genes between i^{th} and j^{th} positions will be decremented by one, i.e. the gene at $(i+1),(i+2)$ positions will be moved to i^{th} and $(i+1)^{th}$ positions, respectively and so on. Similarly, the gene at i^{th} position will be moved to j^{th} position and the gene associated at j^{th} position should be moved to $(j-1)^{th}$ position. For the second part of the parent chromosome that constrains the total cities allotted for each salesman, the genes will be moved as usual to produce new offspring. Examples for Swap, Reverse swap/Flip and Slide operations are illustrated in Figs. 6-8, respectively.

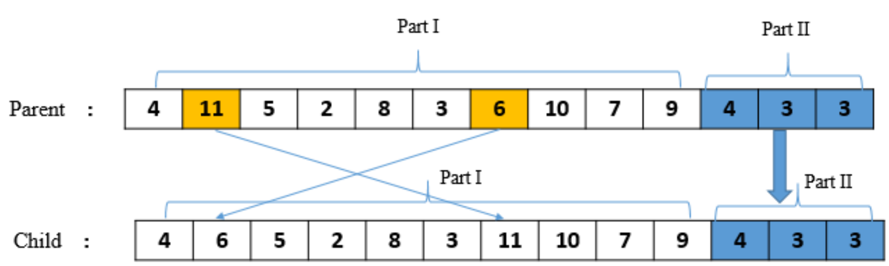


Fig 6. Swap Operator

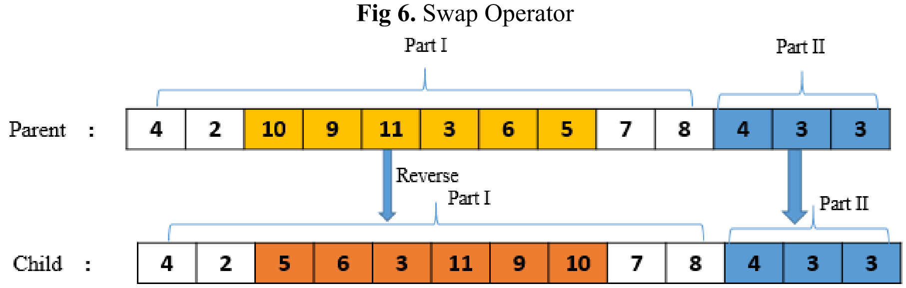


Fig 7. Reverse Swap Operator

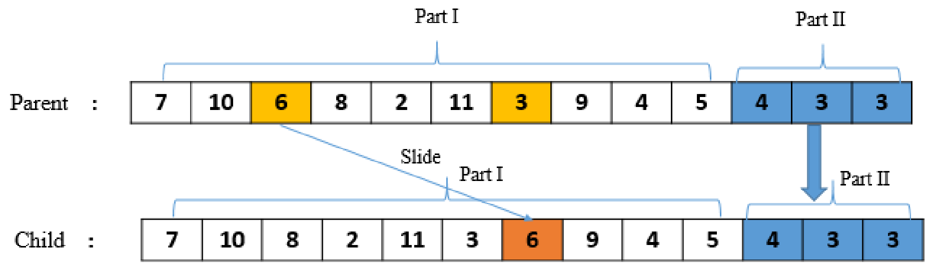


Fig 8. Slide Operator

3.1.7. GA Parameters

In addition to the key elements of GA discussed earlier, setting appropriate values to the parameters namely, size of the population, crossover probability rate, mutation probability rate and termination criteria also plays a vital role in the algorithm's efficiency. These parameter values are based on the problem to be tackled. The Population size indicates the number of chromosomes in any one generation and in this study, it is considered sufficient as large as 100. Crossover probability rate (P_c) tells how often a crossover operation will be executed. If no crossover operation is performed, more chances that produced offspring become a duplicate copy of the parent chromosome. When the crossover takes place, offspring are made with partial features of the parent chromosome. In this study, the crossover probability rate is fixed as 0.85. Mutation probability rate (P_m) indicates how frequently the mutation operation is performed to the parts of the chromosome. This operator makes changes in the part of the chromosomes and thus maintains the diversity in the population. Generally, Pm lies between 0.001 and 0.1. To our study, it is considered as 0.01. Finally, the termination criterion of the GA is assumed to be a maximum number of generations.

4. Computational Results

In this section, the computational results are presented to assess the efficiency of our proposed GA. For all experiments, the GA approach uses roulette wheel selection, modified distance-preserving crossover, complex mutation (Swap, Reverse Swap and Slide) operators, to produce new offspring in every generation. The proposed GA is coded in MATLAB R2017a on a PC with Intel Core CPU i3 2.00 GHz and 4GB of Ram with Windows 10 Pro 64 bit operating system. Unlike the conventional TSP, no open benchmark instance is available to test OMTSPLB. Hence, the benchmark instances available in TSPLIB have been used. As there is no existing study on OMTSPLB, the comparison between the algorithms is not carried out. However, a comparative study on the relaxed version of the present model known as classical MTSP is performed. To assess the efficiency of proposed GA, a comparative study of our GA against the best-known results (BKS) and the new crossover and population generation based Genetic algorithm (GA*) proposed by Singh et al. (2018) for MTSP is performed. The comparative results on twelve-benchmark instances of Carter & Ragsdale, (2006) is given in Table 1. The twelve benchmark instances including MTSP-51, MTSP-100, and MTSP-150 of size 51, 100, and 150 cities, respectively with a distinct number of salesmen are summarized in Table 1. For each instance, the average (Avg.), worst, and the best-found results over 10 independent runs are reported and compared with the best-known results of MTSP. From the results, it is seen that our GA provides best results than the results of GA* for all the twelve instances. It is also evident that the worst solutions of our GA are better than the results of GA* for all the instances. Further, the proposed GA results coincide with the best-known results (BKS) for 9 instances (i.e. 1, 2, 3, 4, 6, 7, 9, 11 and 12). However, for the rest of the instances, the GA finds close results to the BKS. To assess the deviation, gap percentage (Gap1 %) is computed for each instance using the formula (10) where *BKS* is the best known solution available in the literature, *best-found solution* is the solution produced by the proposed GA for a particular instance. The Gap percentage computed is as high as -3.03%. A negative value of gap percentage represents the BKS is better than the best-found solution. Further, the gap percentage (Gap2 %) is measured between BKS and GA* using the formula (11). The Gap percentage computed is as high as -75.09%. Figures 9-11 graphically compare the BKS, GA* and proposed GA on instances of Carter and Ragsdale, (2006). The overall results have shown that the proposed GA is efficient and competent in solving MTSP. Furthermore, the OMTSPLB is solved on the same test instances and the best-found results are reported in column 9 of Table 1. It is seen that most of the best-found results of OMTSPLB are less than the BKS of MTSP. It is due to the number of edges present in OMTSPLB solution is less than the edges present in MTSP solution. From the results; it is evident that, as the number of salesmen increases, the overall distance has a trend of decreasing and the same has been demonstrated in Figs. 12-14.

$$Gap1\% = \frac{BKS - Best\,found\ solution}{BKS} \times 100 \tag{10}$$

$$Gap2\% = \frac{BKS - GA^*}{BKS} \times 100 \tag{11}$$

To test the efficiency of OMTSPLB, a set of 19 benchmark instances have been considered. All these are all Euclidean distances, from which the symmetric distance matrix will be generated. Each instance with a distinct salesman (i.e.

m=2,…,10), and overall, 171 cases have been tested. The best-found results of OMTSPLB are presented in Table 2. The column *Instance* indicates the labels of the benchmark instances; column *Size* lists the size of the tested instances. The proposed GA solves all the given benchmark instances with the distinct salesman varying from 2 to 10. As presented in Table 2, it is seen that the results produced differ for each case. However, the obtained results are varied for each instance; there is no specific rule to find how many salesmen have to be used to get the best result. For instance, the result of *gr96* tried using 4 salesmen is the best of all the presented solutions for this benchmark instance. Similarly, of all the results of the instance *rd100*, the best one is the result corresponding to 10 salesmen. It is, however, a certainty that the topology of the particular problem influences the computational result with distinct numbers of salesmen. In addition, the last column represents the mean computational time required to solve each instance with a distinct quantity of salesman. The mean time (in seconds) shows that the proposed GA finds the best results quickly.

Table 1
Comparative results of the proposed GA with BKS and existing GA on Carter, & Ragsdale, (2006) Instances

Instance	n	m	BKS	GA*	Proposed GA			OMTSPLB	Gap1 %	Gap2 %
					Avg.	Worst	Best	Solution		
mtsp-51	51	3	424	460	428	446	424	413	0	-8.49
	51	5	460	499	463	474	460	439	0	-7.17
	51	10	568	669	586	602	568	468	0	-17.78
mtsp-100	100	3	21,472	22,959	21648	22,091	21,472	22053	0	-6.92
	100	5	23,073	24,559	23,251	23,768	23,182	22448	-0.47	-6.44
	100	10	26,961	33,136	27,144	27,596	26,961	22587	0	-22.90
	100	20	38,245	62,963	38,396	42,502	38,245	22998	0	-64.63
nmtsp-150	150	3	29,390	39,504	34,997	38,208	30,281	39953	-3.03	-34.41
	150	5	30,308	39,862	31,506	32,971	30,308	39993	0	-31.52
	150	10	35,510	50,892	36,633	40,898	35,802	40715	-0.82	-43.31
	150	20	44,697	77,668	46,866	47,182	44,697	42886	0	-73.76
	150	30	58,757	102880	61,428	69,361	58,757	43841	0	-75.09

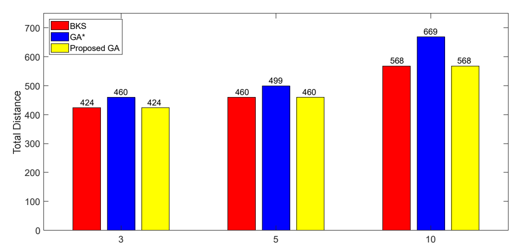


Fig 9. Comparison of minimum distance on mtsp-51

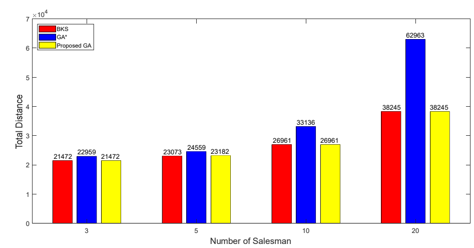


Fig 10. Comparison of minimum distance on mtsp-100

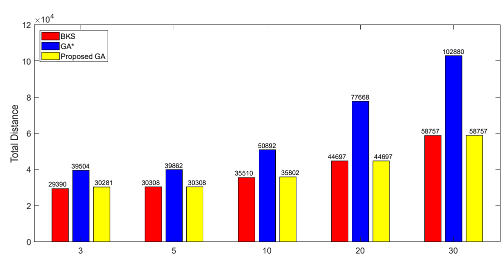


Fig 11. Comparison of minimum distance on mtsp-150

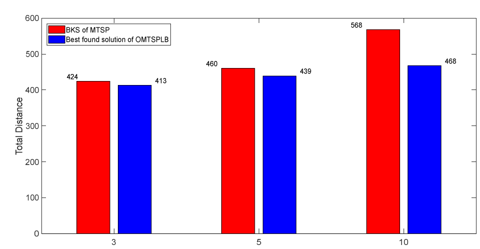


Fig 12. BKS of MTSP Vs. best found solution of OMTSPLB on mtsp-51

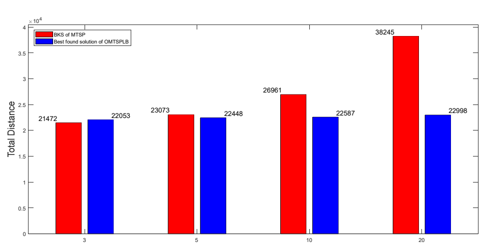


Fig 13. BKS of MTSP Vs. best found solution of OMTSPLB on mtsp-100

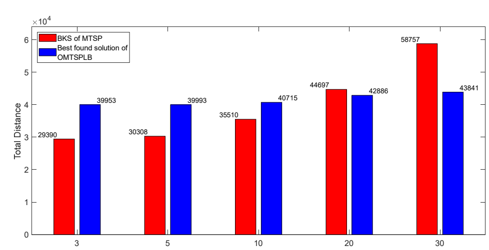


Fig 14. BKS of MTSP Vs. best found solution of OMTSPLB on mtsp-150

Table 2
Computational results of OMTSPLB

Instance	n	Number of Salesman									T
		m = 2	m = 3	m = 4	m = 5	m = 6	m = 7	m = 8	m = 9	m = 10	
att48	48	31873	32931	32830	34750	34830	35472	35292	36743	37510	21
eil51	51	419	432	447	442	458	469	470	483	466	26
berlin52	52	7364	7542	7501	7940	7828	7941	7937	7910	7985	28
st70	70	712	723	689	682	708	695	701	721	705	34
eil76	76	557	560	567	579	572	567	584	593	608	37
pr76	76	112301	114728	112966	115899	119892	121205	121559	121432	125361	41
gr96	96	533	567	526	559	566	560	579	573	564	44
rat99	99	1335	1338	1338	1324	1322	1372	1370	1399	1421	42
rd100	100	8447	8290	8404	8461	8521	8554	8398	8533	8169	56
kroA100	100	23219	23630	22931	23676	23973	23376	23496	23417	24304	54
kroB100	100	23517	23932	23240	23240	23426	23818	23788	23910	23852	58
kroC100	100	21798	21237	21826	22098	22305	22332	22589	21468	22273	54
kroD100	100	22100	22320	22745	23141	22625	23015	23423	22647	22890	61
kroE100	100	22958	23197	23288	22801	23221	23206	23269	23332	23245	62
lin105	105	15816	15447	15752	15206	15262	14889	15802	15523	15504	82
pr107	107	37725	44164	42986	42259	41997	41572	41078	41534	41784	80
pr124	124	59359	57594	63594	63634	62642	60278	62537	62630	63468	114
bier127	127	121715	119617	119030	118800	120351	118321	119576	121445	120554	126
Ch130	130	6416	6492	6524	6472	6626	6735	6696	6731	6564	131

5. Conclusion

This paper addresses a novel variant of MTSP called an *open multiple travelling salesman problem with load balancing constraint (OMTSPLB)*, which finds a wide range of applications in outsourcing logistics distribution and transportation. The objective of this problem is to determine a set of m Hamiltonian paths for m salesman with an overall minimum distance/cost subject to the load balance. A new two-part chromosome representation based crossover and mixed strategy mutation (Swap, Reverse Swap/Flip and Slide) operator is used in the genetic algorithm (GA) for solving OMTSPLB. Proposed GA of complex mutation provides a wide variability in the population and has a less possibility of occurring redundant members in the search space. The comparative results have shown that the proposed GA is efficient and competent in solving classical MTSP. The overall computational results showed that the proposed GA certainly well addresses the number of cities assigned to each salesman as well as minimizes the total traversal distance/cost. Further, the best-found solutions for a set of 19 benchmark instances with distinct salesmen $(\text{for } m = 2,3,..10)$ have been provided, which may be used for future comparative study.

References

Al-Omeer, M. A., & Ahmed, Z. H. (2019, April). Comparative study of crossover operators for the MTSP. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-6). IEEE.

Alves, R. M., & Lopes, C. R. (2015, May). Using genetic algorithms to minimize the distance and balance the routes for the multiple traveling salesman problem. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3171-3178). IEEE.

Bailey, J. D. (1967), The behaviour of adaptive systems which employ genetic and correlation algorithms, Ph.D thesis, University of Michigan.

Benavent, E., & Martínez, A. (2013). Multi-depot Multiple TSP: a polyhedral study and computational results. *Annals of Operations Research*, *207*(1), 7-25.

Bhavani, V., & Murthy, M. S. (2006). Truncated M-travelling salesmen problem. *Opsearch*, *43*(2), 152-177.

Bharath-Kumar, K., & Jaffe, J. (1983). Routing to multiple destinations in computer networks. *IEEE Transactions on Communications, 31*(3), 343-351.

Brown, E. C., Ragsdale, C. T., & Carter, A. E. (2007). A grouping genetic algorithm for the multiple traveling salesperson problem. *International Journal of Information Technology & Decision Making*, *6*(02), 333-347.

Carter, A. E., & Ragsdale, C. T. (2002). Scheduling pre-printed newspaper advertising inserts using genetic algorithms. *Omega*, *30*(6), 415-421.

Carter, A. E., & Ragsdale, C. T. (2006). A new approach to solving the multiple traveling salesperson problem using genetic algorithms. *European journal of operational research*, *175*(1), 245-257.

Garey, M.R., & Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness, in computers and intractability, vol. 24, New York, pp 90–91

Ghadiry, W., Habibi, J., & Aghdam, A.G. (2015) Generalized formulation for trajectory optimization in patrolling problems. In: Proceedings Halifax, NS, CCECE, pp 231–236

Gorenstein, S. (1970). Printing press scheduling for multi-edition periodicals. *Management Science*, *16*(6), B-373.

Harrath, Y., Salman, A. F., Alqaddoumi, A., Hasan, H., & Radhi, A. (2019). A novel hybrid approach for solving the multiple traveling salesmen problem. *Arab Journal of Basic and Applied Sciences*, *26*(1), 103-112.

Hong, S., & Padberg, M. W. (1977). A note on the symmetric multiple traveling salesman problem with fixed charges. *Operations Research*, *25*(5), 871-874.
Kaliaperumal, R., Ramalingam, A., & Sripriya, J. (2015, March). A modified Two part chromosome Crossover for solving MTSP using Genetic algorithms. In *Proceedings ICARCSET 2015, New York,* pp. 1-4.
Kim, K. H., & Park, Y. M. (2004). A crane scheduling method for port container terminals. *European Journal of operational research*, *156*(3), 752-768.
Király, A., & Abonyi, J. (2011). Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm. In *Intelligent Computational Optimization in Engineering* Vol. 366, pp. 241-269.
Larki, H., & Yousefikhoshbakht, M. (2014). Solving the multiple traveling salesman problem by a novel meta-heuristic algorithm. *Journal of Optimization in Industrial Engineering*, *7*(16), 55-63.
Liu, M, & Zhang, PY (2014) New hybrid genetic algorithm for solving the multiple traveling salesman problem: an example of distribution of emergence materials. J Syst Manag 23(02):247–254.
Lo, K. M., Yi, W. Y., Wong, P. K., Leung, K. S., Leung, Y., & Mak, S. T. (2018). A genetic algorithm with new local operators for multiple traveling salesman problems. *International Journal of Computational Intelligence Systems*, *11*(1), 692-705.
Malmborg, C. J. (1996). A genetic algorithm for service level based vehicle scheduling. *European Journal of Operational Research*, *93*(1), 121-134.
Modares, A., Somhom, S., & Enkawa, T. (1999). A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems. *International Transactions in Operational Research*, *6*(6), 591-606.
Okonjo-Adigwe, C. (1988). An effective method of balancing the workload amongst salesmen. *Omega*, *16*(2), 159-163.
Saleh, H. A., & Chelouah, R. (2004). The design of the global navigation satellite system surveying networks using genetic algorithms. *Engineering Applications of Artificial Intelligence*, *17*(1), 111-122.
Sedighpour, M., Yousefikhoshbakht, M., & Mahmoodi Darani, N. (2012). An effective genetic algorithm for solving the multiple traveling salesman problem. *Journal of Optimization in Industrial Engineering*, (8), 73-79.
Shuai, Y., Yunfeng, S., & Kai, Z. (2019). An effective method for solving multiple travelling salesman problem based on NSGA-II. *Systems Science & Control Engineering*, *7*(2), 108-116.
Singh, A., & Baghel, A. S. (2009). A new grouping genetic algorithm approach to the multiple traveling salesperson problem. *Soft Computing*, *13*(1), 95-101.
Singh, D. R., Singh, M. K., Singh, T., & Prasad, R. (2018). Genetic algorithm for solving multiple traveling salesmen problem using a new crossover and population generation. *Computación y Sistemas*, *22*(2).
Sofge, D., Schultz, A., & De Jong, K. (2002, April). Evolutionary computational approaches to solving the multiple traveling salesman problem using a neighborhood attractor schema. In *Workshops on Applications of Evolutionary Computation* (pp. 153-162). Springer, Berlin, Heidelberg.
Tang, L., Liu, J., Rong, A., & Yang, Z. (2000). A multiple traveling salesman problem model for hot rolling scheduling in Shanghai Baoshan Iron & Steel Complex. *European Journal of Operational Research*, *124*(2), 267-282.
TSPLIB: The Library of TSP Benchmark Instances: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
Thenepalle, J., & Singamsetty, P. (2019). An open close multiple travelling salesman problem with single depot. *Decision Science Letters*, *8*(2), 121-136.
Wang, X., & Regan, A. C. (2002). Local truckload pickup and delivery with hard time window constraints. *Transportation Research Part B: Methodological*, *36*(2), 97-112.
Xu, X., Yuan, H., Liptrott, M., & Trovati, M. (2018). Two phase heuristic algorithm for the multiple-travelling salesman problem. *Soft Computing*, *22*(19), 6567-6581.
Yan, X., Zhang, C., Luo, W., Li, W., Chen, W., & Liu, H. (2012). Solve traveling salesman problem using particle swarm optimization algorithm. *International Journal of Computer Science Issues (IJCSI)*, *9*(6), 264.
Yousefikhoshbakht, M., Didehvar, F., & Rahmati, F. (2013). Modification of the ant colony optimization for solving the multiple traveling salesman problem. *Romanian Journal of Information Science and Technology*, *16*(1), 65-80.
Yuan, S., Skinner, B., Huang, S., & Liu, D. (2013). A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms. *European Journal of Operational Research*, *228*(1), 72-82.