Alexandria Engineering Journal 90 (2024) 98–111

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Genetic algorithm to the bi-objective multiple travelling salesman problem

Shayathri Linganathan, Purusotham Singamsetty *

Department of Mathematics, School of Advanced Sciences, Vellore Institute Technology, Vellore, India

ARTICLE INFO

Keywords:
Travelling salesman problem
Multiple travelling salesman problem
Genetic algorithm with tournament selection

ABSTRACT

The travelling salesman problem (TSP) and its variants have been studied extensively due to its wide range of real-world applications, yet there are challenges in providing efficient algorithms to deal with some of its variants. The multiple travelling salesman problem (MTSP), is the generalization of TSP, which aims to determine $m-$ routes for 'm' salesmen to cover a set of $n-$ cities exactly once where each route starts and ends at a depot such that the total distance is least. In this, the number of cities in each route of the optimal solution may be distributed disproportionately. This paper presents, a bi-objective MTSP (BMTSP) with the load balancing constraint, where the first objective is to minimize the total travel distance and the second objective minimizes the total time. A metaheuristic based genetic algorithm with tournament selection (GATS) is designed by integrating with mixed strategies, such as flip, swap and scramble in mutation operation to obtain efficient Pareto solution for BMTSP. The computational experiments are carried out on different data sets, which are derived from the TSPLIB. The performance of GATS is compared with different genetic approaches and simulation results show that the proposed GATS obtained improved solutions on some of the benchmark instances.

1. Introduction

TSP is typically a large class of hard optimization problems. Also, the problem is used as the benchmark for many optimization methods. It is one of the widely studied NP-hard problems in combinatorial optimization. In TSP, a single salesman has been involved and aims to visit all the $n-$ cities exactly once, starting and ending at a depot, with the least traversal cost. In general, when the number of cities increases the computational complexity increases exponentially and it is unrealistic to visit all of them with a single salesman. Thus, it is to be anticipated to make use of multiple salesmen to cover all the cities. However, TSP and its variants have been extensively studied by several researchers and focused on problem-solving in various models and algorithmic approaches, including exact methods such as branch and bound, Lexi search, branch-and-cut, branch-and-price, etc., heuristics, and metaheuristics such as Genetic Algorithms (GA), Simulated Annealing (SA), Tabu Search (TS), Particle Swam Optimization (PSO), Artificial Neural Networks (ANN), Ant Colony Optimization (ACO), farmland fertility algorithm (FFA) [1,21], harris hawk optimization algorithm [20], etc. Also, the enlargement of TSP models gives many applications such as vehicle routing problems [54], maritime transportation [40], logistics distribution and transportation [37], School bus routing problem (Angel, et.al. 1972), manufacturing ([34]), matching's [31], etc.

In MTSP, there are m-salesmen to visit all $n(>m)$ cities and all of them positioned at a single depot in an $n-$ city network. The MTSP seeks m-tours by partitioning the $n-$ cities into m subsets, only a depot city is common in all the subsets, such that each subset of cites is covered by exactly one salesman. The tour of each salesman starts from the depot, after visiting a subset of cities exactly once, returns to the depot, and the total travel distance covered by all the salesmen should be the least. TSP is the special case of MTSP for $m=1$, known to be NP-hard. MTSP is much harder than TSP, as it involves a selection plan (i.e. which cities to be visited by a salesman) and an optimal tour plan (i.e. the ordering of cities within the allotted cities) for each salesman.

Several variants of MTSP's are emerged such as symmetric and asymmetric MTSP [6,26], MTSP with multiple depots [27,55], truncated MTSP in which a subset of cities will be covered with minimum distance by m-salesmen [11], open-closed MTSP, the tour begins and ends at a single depot called closed tour, where as an open tour begins at depot but does not end at the depot [29], MTSP with load balancing [43], MTSP with time windows, deals with finding a set of optimal routes for a fleet of vehicles in order to serve a set of locations, each one within a specified time window [49], the multi-agent planning problem, in which there are n-targets or goals and the objective is to find m-tours such that each target is visited by exactly one agent once and the total cost of visiting all such targets is minimum [52], bi-objective or multi-objective

* Corresponding author.
E-mail address: drpurusotham.or@gmail.com (P. Singamsetty).

https://doi.org/10.1016/j.aej.2024.01.048
Received 23 September 2023; Received in revised form 29 December 2023; Accepted 14 January 2024
Available online 31 January 2024

MTSP, etc. Alok and Anurag [7], studied an MTSP variant, in which they considered two alternatives one minimizes the total distance travelled by all the m-salesmen, and the other minimizes the maximum distance travelled by any one salesman, which helps to balance the load or distance travelled among the salesmen.

The MTSP variants are more appropriate than the usual TSP for many practical problems arising in network routing, e-commerce supply and distribution, vehicle routing and scheduling problems, etc. To mention, the MTSP is applied in job scheduling that includes different parallel production units in print press [14], security services [13], hot roll scheduling in Shanghai Baoshan Iron & Steel Complex [53], school bus routing that looks for planning of bus routes with minimal distance subject to no bus is overcrowded and without violating the threshold on the maximum allowable time [10], pickup and delivery (Wang and Regan, 2002), oil tankers dispatching problem [2], workforce planning and workload balancing [42]. The vehicle scheduling problem (VSP) can also be viewed as the MTSP [15], centralized and distributed path planning for multi-agent exploration [24], LPG distribution path planning in rural areas [30], etc.

The MTSP is a very hard combinatorial optimization problem, so obtaining its optimal solution using exact algorithms is very difficult and sometimes not possible. Therefore, the researchers are interested in developing better heuristic and metaheuristic based algorithms to find near optimal solutions within an affordable computational time, instead of looking for an exact optimal solution. Russell [45] suggested an mTOUR heuristic, which includes two stages: clustering of nodes first – touring second. Laporte and Nobert [35] solved MTSP instances sizes up to 100 cities with a cutting planes algorithm. Gavish and Srikanth [23] proposed an efficient branch-and-bound algorithm to solve MTSP optimally, where the lower bounds are calculated using the Lagrangian relaxation technique that is integrated with a degree-constrained minimal spanning tree. Song et al. [50] developed an extended simulated annealing algorithm, and this gives rise to perturbation schemes that correspond to the problem-augmented TSP or MTSP. To equally distribute the workload for the salesmen, this algorithm was incorporated with entropy constraints along with energy function. [32], designed an ant colony optimization scheme for the MTSP, tested on some standard data sets available in TSPLIB and obtained comparable solutions.

In recent one-two decades several forms of GA's evolved and have been implemented successfully on MTSP variants. Sze and Tiong [51] provided a comparative study on the computational run time between the nearest neighbour algorithm (NNA) and GA for solving the MTSP, and the results show that NNA is superior to the conventional GA. Brown et al. [12] proposed a grouping GA designed specifically for grouping or clustering problems such as MTSP. This technique introduces a new chromosome representation to indicate the tour of a sequence of cities and which salesman is assigned to each tour. The computational results outperform the traditional encodings of other previously existing GA methods.

Zhou and Li [65] solved an MTSP problem using GA, in which the initial solution is generated with the help of a greedy search algorithm, obtained the effective neighbouring solutions by performing the mutation operation with 2-Opt local search strategy to move towards an optimal solution. Chen and Chen [17] propose a genetic algorithm in which a two-part chromosome encoding strategy is employed in mutation and recombination operators, giving the solution of the MTSP. This two-part chromosome encoding takes care of the ordering of cities and the number of cities to be visited by each salesman. Sedighpour et al., [46] proposed a modified GA, used a 2-Opt local search algorithm for improving the solutions. The efficiency of the algorithm was tested on various TSPLIB test instances.

An MTSP is solved in two stages by Yousefikhoshbakh and Seidighpour [59], in which a sweep algorithm is used in the first stage and second stage an ant colony optimization technique with 3-Opt local search strategy is used to improve the solutions obtained in the first stage. [44], developed a modified GA to solve the MTSP, generate an effective initial population and investigate the search space quickly with the help of local search operators to find promising solutions. The computational details on some benchmark data sets were provided and the improved solutions on the instances ranging between 76 and 1002 cities when compared with the results reported by Yousefikhoshbakh and Seidighpour [59] and Junjie and Dingwei [32]. Youssef et al. [61] developed a hybrid algorithm for MTSP, which is a combination of modified ant colony, 2-Opt local search and genetic algorithm. In this, an ant colony scheme is used to generate the solutions followed 2-Opt mechanism is applied to enhance the solutions and finally genetic algorithm is applied to find the best possible solutions. The comparative results provided on selected TSPLIB benchmark instances improved solutions on small and medium-sized data sets.

Although MTSP is studied extensively, the study on bi-objective or multi-objective MTSP is limited. Venkatesh and Singh [56] considered a bi-objective MTSP, in which the first one is to minimize the total distance covered by all the salesmen and the second objective, is to minimize the maximum distance covered by a salesman. This second objective in turn helps to balance the distance travelled among the salesmen. Two heuristic methods are used to solve this bi-objective MTSP, where the first one is an artificial bee colony algorithm, and the second one is an invasive weed optimization algorithm. In addition, a local search is used to improve the solutions obtained in previous schemes. The computational results show some prominent solutions on some benchmark instances. Shuai et al. [47] addressed a bi-objective MTSP, in which the first objective is to minimize the total distance covered by the m-salesman and the second objective is to minimize the difference between longest route and shortest route. An efficient multi-objective evolutionary algorithm is used to find the Pareto solutions. The crossover and mutation operators are designed with NSGA-II intelligently, which helps to jump from the local optima and hence to find the best solutions. The experimental results are compared with several state-of-the-art algorithms. Purusotham and Jayanth Kumar [43] propose an efficient genetic algorithm to solve an open MTSP with a load balancing constraint. This problem differs from the conventional MTSP, where all salesmen start at the depot and visit not more than the specified number of cities without returning to the depot.

Youssef [60] considered a multiple-objective MTSP. In this there are two parametric objectives, the first one is to balance the workload among the salesman so that they execute the tasks with nearly equal total work (i.e. the best solutions are those that give us nearly equal to the average processing time) and the second one is to minimize the total distance (time) covered by all the salesman together. This problem has some applications in different areas such as scheduling environment, food industry and maintenance activities, etc. A three-phase metaheuristic algorithm was developed to solve this MTSP. The principle of center of mass and a neighbourhood search schemes are used in the first two phases to assign the n-cities to m-salesmen. A TSP solver is used in the third phase to find the optimal tour of each salesman within the assigned cities in phase 2. The computational details show that this metaheuristic outperforms than the clustering algorithm for the load balancing objective. Al-Taani and Al-Afifi [4] used a memetic algorithm to solve the MTSP. More recently, He and Hao [27], formulated a minmax MTSP, in which a memetic search algorithm is developed to determine a tour that minimizes the longest tour among the m-tours. Kloster et al. [33] studied an MTSP with Drone Stations and suggested a decomposition-based metaheuristic algorithm that integrates the local searches. [36] proposed a deep- convolutional neural network approach to solve the MTSP. Table 1, provides the brief summary of the related works on MTSP to the proposed study.

Most of the earlier works on MTSP cited in the literature are formulated on a single independent attribute. However, many real world problem formulations involve two or more independent attributes. This paper presents a BMTSP with the load balancing constraint, which includes the two independent objective parameters namely time and dis-

Table 1
Summarizing the most related works on MTSP to the current study.

Article	Depot (s)	Number of Independent attributes	Objective function type	LBC	Solution method
Proposed work	Single	Two (Time and distance)	Min. distance and Min. time	Yes	GATS
[5]	Single	One	Min.	No	Bees algorithm
[27]	Single and multiple	One	Minmax.	No	Memetic search
[36]	Single	One	Min.	No	Convolutional neural network
[63]	Single	One	Minsum. and Minmax.	No	Two-stage heuristic
[22]	Multiple	One	Min.	No	Firefly algorithm and ACO
[4]	Single	One	Min.	No	Memetic algorithm using GA and HC
[19]	Single	One	Min.	No	K-means clustering, GA and ACO
[43]	Single	One	Min.	Yes	GA
[26]	Single	One	Min	No	Clustering algorithm
[60]	Single	Two (Time and distance)	Min. deviation and Min. distance	Yes	Three-step metaheuristic
[47]	Single	One (distance)	Min. distance and Min. deviation	Yes	Evolutionary algorithm
[39]	Single	One	Min.	No	GA
[56]	Single	One (distance)	Min. and Minmax	Yes	Artificial bee colony algorithm
[7]	Single	One	Min.	No	GA
[12]	Single	One	Min.	No	GA
[15]	Single	One	Min.	No	GA

Min. – Minimizing the total distance/cost; LBC –Load balance constraint

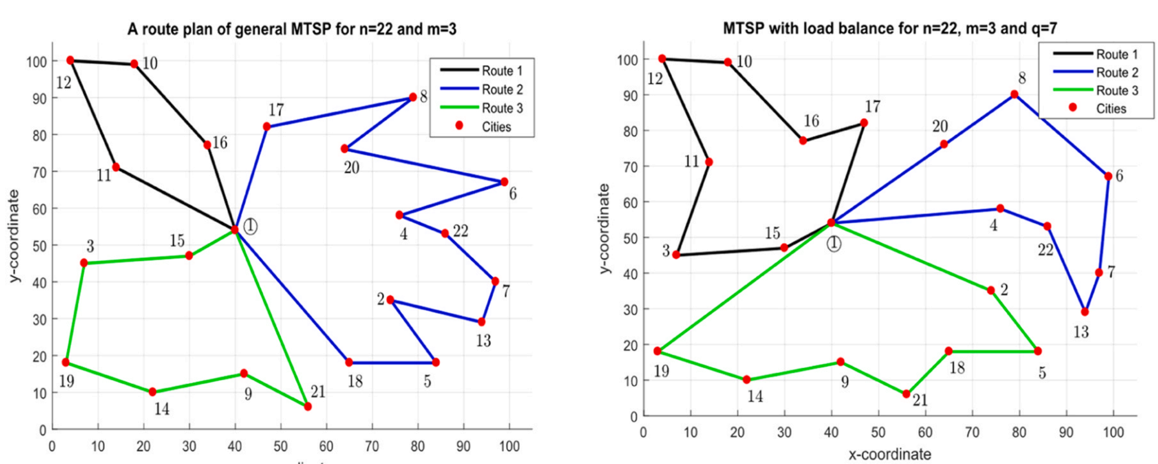
tance. Therefore, the proposed BMTSP could be an extended version of the usual MTSP. This study contributes a metaheuristic based genetic algorithm with tournament selection (GATS), which is designed by integrating with mixed strategies, such as flip, swap and scramble in mutation operation to obtain efficient Pareto solutions for this extended version of BMTSP. Fig. 1(a) illustrates one of the feasible solutions of MTSP for 3-salesman with a single depot on a 22-city network. In which, the coordinates (x_i, y_i) of city locations on the Euclidian plane are randomly generated. Observe that, route-1 consists of 5-cities, route-2 includes 12-cities, whereas in route-3 there are 7 cities, where the depot city-1 is common in all the routes. This type of solution appears to be the disproportionate distribution of cities among the salesmen (i.e. the load of visiting the cities is higher on some salesmen). Fig. 1(b) shows an arbitrary trip schedule of MTSP with load balance in which each salesman is touring 7-cities excluding the depot city; that is, the salesmen are sharing $(n-1)$ cities, excluding the depot, with approximately an equal number to balance the load while touring the cities. This problem has many real-world applications, as explained earlier. The next section includes a detailed description of BMTSP along with its mathematical notion of 0–1 integer programming. Section 3 describes the basic steps in GA and it's flow-chart. The concept of BMTSP is explained in detail through an example in Section 4. The extensive computational details are presented in Section 5. Finally, the concluding remarks are summarized in Section 6 and followed a list of references is provided.

2. Problem description and mathematical representation

The Table 2 gives a list of notations used to describe the BMTSP:

Table 2
Notations.

Notations	Description
n	Number of cities/locations
m	Number of salesmen $(m < n)$
α	A depot city
V	Vertex set
E	Edge set $\{e_{ij} = (i,j) \mid i,j \in V\}$
D	Set of distance values of $d_{ij}, i,j = 1,2,\ldots,n$
d_{ij}	The distance (Euclidean distance) between a pair of cities i and j
T	Set of time values of $t_{ij}, i,j = 1,2,\ldots,n$
t_{ij}	Time (travel time from city i to city j + the amount of time spent to execute the assigned tasks at city j).
q	Each salesman must visit at most q-cities excluding depot city, where $q = \left\lceil \frac{n-1}{m} \right\rceil$ ($\lceil x \rceil$- denotes the least integer greater than or equal to x)
S_k	A subset of cities that are visited by a salesman k $(1 \le \lvert S_k \rvert \le q+1,$ $k = 1,2,\ldots,m)$ and a depot city α only common in all these subsets
δ_{ijk}, y_{ik}	Binary decision variables



(a) A route plan of a general MTSP (b) A route plan of MTSP with load balance

Fig. 1. (a) A route plan of a general MTSP. (b) A route plan of MTSP with load balance.

Let $G(V, E, D, T)$ denote a complete weighted graph, where V and E respectively denote, the vertex set representing n-cities and the edge set which connects the cities in V. The distance between each pair of cities d_{ij} is known. In general, $d_{ij} > 0$ and $d_{ij} = \infty$ if the cities i and j are not directly connected. Also if $d_{ij} = d_{ji}$ then the graph is known to be an undirected graph and the matrix D becomes symmetric, if $d_{ij} \neq d_{ji}$, then the graph is directed and D will be an asymmetric matrix. Let $\alpha \in V$ be the depot city. Let there are m-salesman, all of them positioned at the depot point. Each salesman starts at the depot, wishes to visit a subset of cities to execute the assigned tasks and returns to the depot. In addition to the distance, the time of travel between a pair of cities i and j together with the processing/execution time of tasks at city j is denoted by t_{ij}. In simple, two parametric values namely distance (d_{ij}) and time (t_{ij}) are given on each edge that connects a pair of cities i and j. Usually, the MTSP aims to determine m-shortest routes for 'm′ salesmen all of which start and end at a depot city; the salesmen should cover all the n-cities without intervening such that the total distance travelled by the salesmen is least. However, minimizing only the total distance will result in highly disproportionate distribution of cities where one salesman may cover most cities whereas each of the other salesmen covers one or a few cities (See Fig. 1(a)).

In this paper, a bi-objective MTSP with load balancing constraint is considered, where the first objective is to minimize the total travel distance and the second objective minimize the total time covered by the salesmen. The travel time between a pair of cities is not directly proportional to the distance between the same cities as it can be influenced by traffic intensity, road convenience, etc. Therefore time and distance are considered as two independent parameters. The problem also consists of a selection plan (i.e. selecting a subset of cities S_k with not more than q-cities from V which are to be visited by a salesman k) and a route plan (i.e. finding a tour with the cities in S_k). Now, the optimal solution to this problem may not be possible at one point as it involves a trade-off between the two objectives. Therefore, one can look for the best Pareto solutions for bi-objective or multi-objective optimization. The mathematical representation of the BMTSP is given below:

$$\text{Minimize} Z = (z_1, z_2) \tag{1}$$

where,

$$\left.\begin{aligned} z_1 &= \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}\delta_{ijk} \\ z_2 &= \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij}\delta_{ijk} \end{aligned}\right\} \tag{2}$$

Subject to the constraints:

$$\sum_{k=1}^{m} \sum_{j=1}^{n} \delta_{\alpha jk} = m \tag{3}$$

$$\sum_{k=1}^{m} \sum_{i=1}^{n} \delta_{i\alpha k} = m \tag{4}$$

$$\sum_{k=1}^{m} \sum_{j=1}^{n} \delta_{ijk} = 1, \forall i \in V \backslash \{\alpha\} \tag{5}$$

$$\sum_{k=1}^{m} \sum_{i=1}^{n} \delta_{ijk} = 1, \forall j \in V \backslash \{\alpha\} \tag{6}$$

$$\sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \delta_{ijk} = m + n - 1 \tag{7}$$

$$\sum_{i \in S_k} \delta_{ipk} - \sum_{j \in S_k} \delta_{pjk} = 0, \forall p \in V, k = 1, 2, \ldots, m \quad + \text{Sub} - \text{tour elimination constraints} \tag{8}$$

$$\sum_{j=1}^{n} y_{ik} \leq q, k = 1, 2, \ldots, m \tag{9}$$

$$\delta_{ijk}, y_{ik} \in \{0, 1\} \tag{10}$$

The Eq. (2) provides the total distance travelled and total time taken to execute all the tasks by the m-salesman along the m-routes, and the goal is to minimize both these parameters as given in Eq. (1). Eqs. (3) and (4) respectively, denote that the m-tours for m-salesmen start and end at depot city α. Each city j except the depot city is to be visited by exactly one salesman. The salesman's arrival and departures are respectively given in Eqs. (5) and (6). The Eq. (7) states that any feasible solution of this BMTSP must contain $m + n - 1$ edges. The Eq. (8) holds the continuity of the trip with each of the selected subset of cities S_k for the salesman k and in addition, any formation of sub-tours will be eliminated. The constraint (9) is imposed to balance the load on the salesmen i.e. the total number of cities visited by a salesman k should not be more than q. Finally, the binary variables given in (10) are defined as follows:

$$\delta_{ijk} = \begin{cases} 1, \text{if city} j \text{is visited from city i by a salesman} k \\ 0, \quad \text{otherwise} \end{cases}$$

and

$$y_{ik} = \begin{cases} 1, \text{if city} i \text{is visited a salesman} k \\ 0, \quad \text{otherwise} \end{cases}.$$

The BMTSP with load balance can be used to formulate many applications arise in industry, food/goods delivery, tourist route planning, humanitarian logistics etc. For example, the home appliances such as washing machines, water purifiers, refrigerators, air conditioners etc. needs periodical or repair maintenance. A company may receive different consumer complaints or regular service requests over the phone/mail from n − different locations. Let there are m − service persons be available in a company to handle the service requests raised in different locations. Assume that a serviceman cannot handle more than a specified number of service requests over a period of time. The company wishes to assign the servicemen to different locations to resolve the service requests smoothly such that the amount of time spent on these requests and the amount of distance covered by the servicemen is least.

Xiaolong et al. [58] imagined an interesting application of MTSP in tourism route planning where a tourist wants to visit a set of n-tourist spots in m-days with minimum travel distance subjected to the limited daily available time and the number of tourist spots to be balanced. In addition, it is assumed that the tourist must start and end his journey at the same point every day. Moreover, the time spent on visiting the distinct tourist spots will be minimized. The problem involves two objectives similar to the proposed BMTSP. This problem is solved with a two-phase heuristic algorithm (TPHA), which is an improved version of the K-means algorithm by clustering the cities visited by the tourist in the first phase and then a route plan algorithm is designed to ascertain the best route through GA combined with roulette wheel selection method in the second phase. The comparative experimental results show that they obtained better results with TPHA than the ant colony algorithm (ACA) and nearest neighbourhood methods (NN).

3. Genetic algorithm with tournament selection

Genetic algorithm (GA) is a widely used computational algorithm for many hard combinatorial optimization models, popularized with the works of John Henry Holland in 1970 s that was designed with the principles of Darwin's biological evolution theory. Goldberg [25] describes the adaptive heuristic search algorithm as a search-based optimization technique, through which he finds multiple approaches to using the same algorithm for a particular problem. Several evolutionary algorithms are evolved to solve NP-hard problems and applied in diversified domains such as computer-aided design, medicine, telecommunication, microelectronic circuit designs, cryptosystems production and planning management, robotics, etc. In recent one to two decades different forms of GAs are evolved such as GA with two-part chromosomes [15,62], GA with new local operators [8,39], partheno-genetic algorithms (PGA) [64], Hybrid GA (Chao [16]), etc. The GA's are proven to be powerful in finding the optimal or near optimal solutions within a reasonable time by eliminating the solutions which are not feasible or useful. However, the ability of the algorithm may become weaker when the search space is too large and the initial population is for away from optimal. The key ingredients of the algorithm are natural reproduction (selection), crossover and mutation.

3.1. Initial population

The solution space is represented by a collection of individuals and an individual is characterized as a string by Holland, [28]. In practical, it is too expensive to investigate the entire search space. Therefore, a part of the search space called the population is to be examined effectively, to determine a better solution. The population is a subset/initialization of all possible solutions to the given problem. It is an initial process for all random or heuristic searches. An individual in a population analogous to a gene represents a feasible solution to the problem. Each gene can be arranged in the form of an ordered indices as cities of length $\mathrm{n}-1$. Here depot city 1 is excluded in the ordered sequence as it is common in all the $\mathrm{m}-$ tours. There can be $(\mathrm{n}-1)!$ ordered permutations with $\mathrm{n}-1$ cites in the population. Each city appears exactly once in the pattern of the gene. A city can occupy any position in the pattern. A tour of the salesman is to be constructed by joining the depot city at both ends of the ordered cities in each block. In BMTSP a gene containing m blocks corresponds to $m-$ salesman with $n-1$ ordered indices and each block includes at most $q=\left\lceil\frac{n-1}{m}\right\rceil$ cities to satisfy the load balance constraint. Here the size of the population is taken as 100 for all the instances undertaken for computational experiments.

For instance, there are $m=3$ salesmen to cover $n=12$ cities, a gene in population be a pattern of an arrangement of ordered indices of cities excluding depot city (say city-1) of length $n-1=11$ with $\mathrm{m}-$ blocks, each block contains at most $q=\left\lceil\frac{11}{3}\right\rceil=\lceil 3.667\rceil=4$ cities as shown in Fig. 2.

3.2. Fitness function

The fitness function assigns a numeric value to each individual's gene in the population, which gives us a feasible solution to the problem [9]. The fitness function is used to evaluate the solution either the objective function is maximizing or minimizing type. After finding the fitness value of the individuals, it is classified as high, medium and low fitness values. The quality of the gene is measured by its fitness value. A gene with the lowest fitness value is qualitative when the objective function is minimization type whereas a gene with the highest fitness value is qualitative when the objective function is maximization type. However, it is difficult to decide the quality of the gene in the case of bi-objective and multi-objective optimization as it involves the trade-off between the objectives. Clearly, the fitness function is always problem dependent.

In BMTSP, the fitness function assigns a pair of values to each gene concerning the two objective functions given in Eq. (1). Fig. 6b and Fig. 6d contain a collection of pairs of solutions that are generated for the numerical example given in Section 4. It is observed that, although both objective functions are of the minimization type, the fitness value of a gene does not guarantee that both z_1 and z_2 arrive at the lowest value due to trade-off between the two objectives. One can select a better fitness value that corresponds to the first objective by relaxing the second objective and vice versa. In another way the decision-maker can choose an appropriate or prominent solution pair by considering acceptable trade-offs between the two objectives or by hierarchy. Among them, choose a gene with a low fitness value for evaluation and use it to find the nearer optimal/sub-optimal solutions.

5	6	4	7	10	2	12	9	3	8	11
Salesman 1				Salesman 2				Salesman 3		

Fig. 2. The pattern of a random gene.

3.3. Natural selection

In the natural selection process, an individual with a higher fitness value will be selected to generate the next generation genes. In this process, individual genes that are selected will be added to the new population without any exit, and on the other hand some of the individual genes will be removed which are not useful to balance the population size. Several techniques are available in the literature to select the secondary or next generation genes such as Tournament selection, Roulette wheel section, Proportionate selection, Ranking based selection, Steady state selection, etc. In the proposed algorithm tournament selection is incorporated to select the next-level generation genes. In tournament selection, p-individual genes will be selected to play a tournament. The fitness value varies with tournament size, the more of players the selection pressure will be high [41] to win in the tournament. Further, continue the search until the best fitness value or winning gene is obtained and then it will be added to the next generation. In Fig. 3, there are 9-individual genes with their own random fitness values. For example, three individuals X, Y and U are selected at random to play a tournament. Since the objective is of the minimization type, the gene Y is selected, having the lowest fitness value among the three selected genes.

3.4. Crossover

The process of crossover consists of combining two individual genes to create a new one. The system gradually uses random selection to crossover to produce better genes. Here, one parent is selected, and one or more offspring are produced using the genetic material of the parents. Different crossover operations are used in the literature, such as one-point crossover, multi-point crossover, uniform crossover, ordered crossover [18], partially mapped crossover [25], sequential construction crossover [3], etc. In the proposed algorithm, a sequential crossover operation is performed to produce child genes and carefully eliminate all the illegal individual genes that are not feasible to the given problem.

3.5. Mutation

The mutation process is used to maintain genetic diversity in the population. Populations are based on fitness and then muted randomly. There are different mutation operations such as flip, swap, inversion and scramble mutations [48], among others. A mixed combination of mutation operations such as flip (Fig. 4a), scramble (Fig. 4c) and swap (Fig. 4b) together incorporated into the proposed algorithm. This mixed strategy helps to find some improved or near-optimal solutions. Flip the cities in the flip mutation under the breaks for the randomly generated routes, as shown in Fig. 4a. In slide mutation, the cities are sliding across

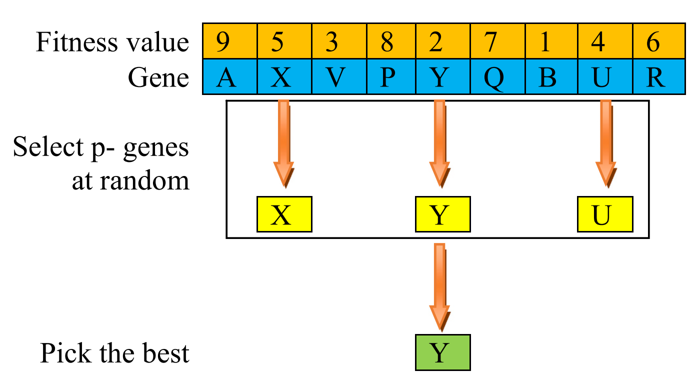


Fig. 3. Tournament selection.

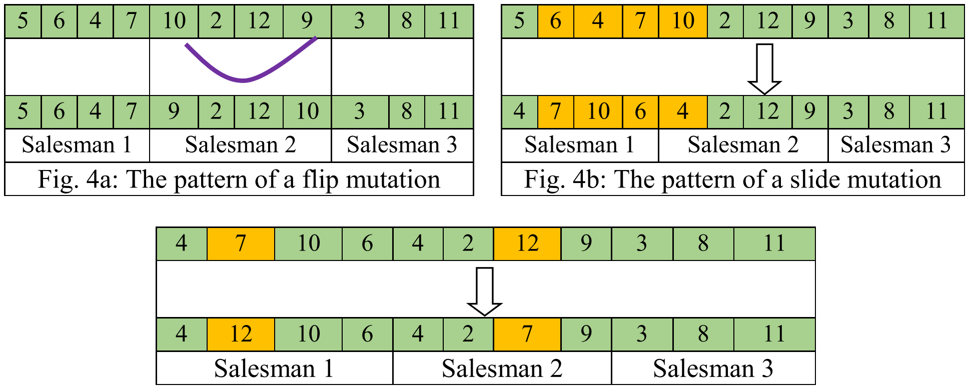


Fig. 4. a The pattern of a flip mutation. b: The pattern of a slide mutation. c: The pattern of a swap mutation.

the routes say $6-4-7-10$ as $7-10-6-4$ to generate a new gene, as shown in Fig. 4b. In swap mutation, select a pair of cities across the gene say $(7,12)$ and swap them as $(12,7)$ to produce a new gene, as shown in Fig. 4c.

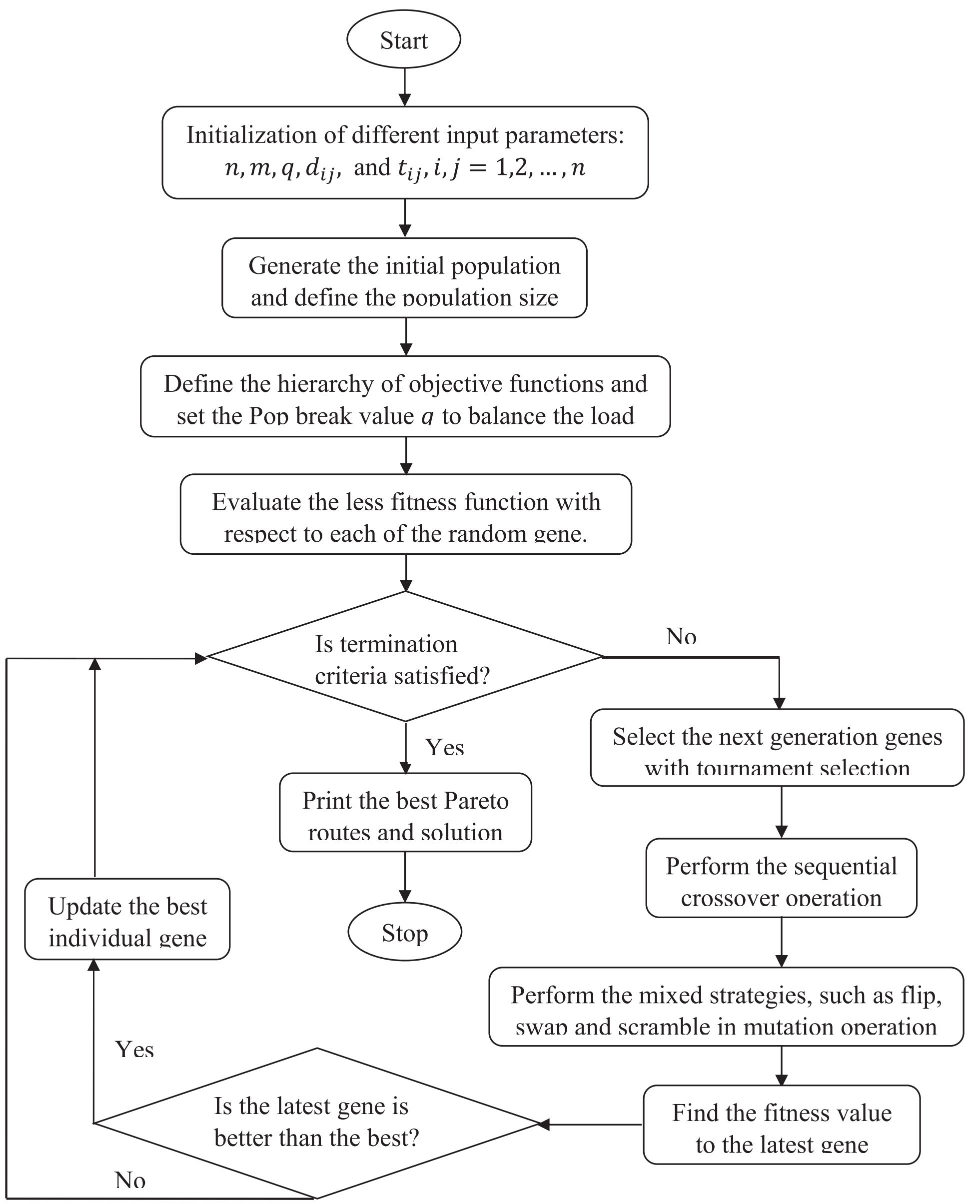


Fig. 5. Flow Chart of the proposed GATS.

3.6. Flow Chart

To determine the best Pareto solutions to BMTSP, the structure of the proposed GATS code implemented in MATLAB, is presented in the form a flow chart in Fig. 5.

4. Numerical illustration

In order to understand the BMTSP and the concepts involved in the proposed algorithm, a numerical example with $n = 22$ cities and $m = 3$ salesmen is considered. Let the coordinates of the 22-cities on the Euclidean plane are given in the set $\{(x_i, y_i) = (40,54), (74,35), (7,45), (76,58), (84,18), (99,67), (97,40), (79,90), (42,15), (18,99), (14,71), (4,100), (94,29), (22,10), (30,47), (34,77), (47,82), (65,18), (3,18), (64,76), (56,6), (86,53)\}$. Let $\alpha = (x_1, y_1)$ be the depot city. The distance between each pair of cities assumes a non-negative quantity. For the first objective, the Euclidean distance $d_{ij}, i, j = 1, 2, \ldots, n$ between a pair of cities $v_i = (x_i, y_i)$ and $v_j = (x_j, y_j)$ is calculated, rounded to the nearest integer value, and used to construct the distance matrix. One can observe that the entries in the distance matrix are symmetric ($d_{ij} = d_{ji}$). For the second objective, the time of travel t_{ij} between a pair of cities v_i and v_j is uniformly generated over an interval $[0, 50]$ using *randi* syntax in MATLAB and is given in Table 3. The entries in Table 3 are not symmetric. Conveniently, the cities $v_i = (x_i, y_i), i = 1, 2, \ldots, n$ are indexed as $1, 2, \ldots, n$ in Table 3. To maintain the load balance among the salesmen, each salesman should select a subset S_k of cities such that $1 \leq |S_k| \leq q + 1$, where $q = \left[\frac{n-1}{m}\right] = \left[\frac{21}{3}\right] = [7] = 7$ cities excluding the depot city to visit. Clearly, the salesman's tour length will be at most $q + 1$. The problem is to determine the $m -$ tours which begin and end at depot city, covers $n -$ cities such that the total distance and total time covered by all the salesmen is least. The salesmen should not intervene in the cities except the depot city while touring. The problem also involves two key aspects; one is which cities are supposed to be visited by a salesman (selection plan) and the other is the ordering of the selected cities (route plan). Note that the problem involves two distinct objectives; the optimal solution cannot be achieved at one point. Therefore, the decision maker can choose an appropriate Pareto solution either within acceptable trade-off between the two objectives or based on the hierarchy (priority) of the objectives.

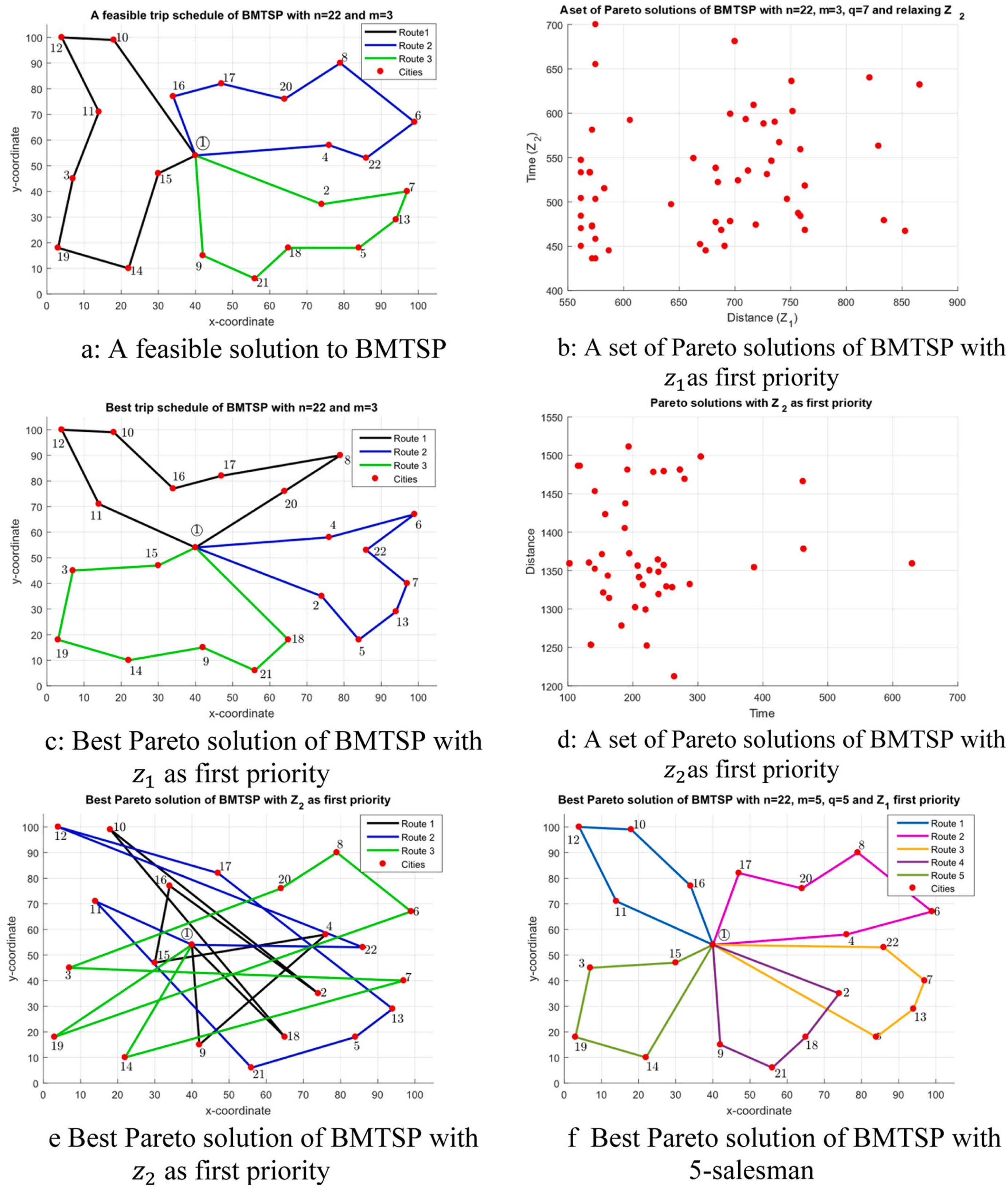
To find the solution to BMTSP, the proposed Genetic Algorithm with Tournament Selection (GATS) is coded in MATLAB environment. Initially, feed the necessary inputs to the algorithmic code, define the population size and hierarchy of objectives (z_1, z_2), then generate the random genes, and finally perform the different operations involved in the algorithm as discussed in Section 3. Conveniently, the first objective (z_1) is assumed to be the first priority, which means finding the m-routes with the minimum total distance for MTSP with load balance and then picking the total minimum time of the second objective z_2 with respect to all alternatives of these m-routes. The solutions obtained on this instance are shown in Fig. 6a–Fig. 6 f.

A feasible solution of BMTSP with $n = 22$, $m = 3$ and $q = 7$ is shown in Fig. 6a, in which the x and y coordinate axes, respectively, denote the locational coordinates of cities. The indices 1, 2, …, 22 denote the city locations on the Euclidean plane. There three route plans for the 3-salesmen to visit the given set of cities. Each salesman starts at depot city 1, traverses through another 7 cities and returns to the depot. All salesmen may be assigned at random to these routes. Further, the direction of the route plans for the three salesmen is given in Table 4. The first salesman takes route 1, travel 173 units of distance in 212 units of time; the second salesman traverses through route 2, covers 179 units of distance in 177 units of time and the third salesman travel 220 units of distance in 192 units of time along route 3. The three salesmen together cover a total of 572 units of distance in 581 units of time while visiting all the cities through these routes.

The scatter plot of a few Pareto solutions generated with the use of GATS is shown in Fig. 6b. In Fig. 6b, the total distance covered by the salesmen is taken on the x-axis and the total time consumed is taken on the y-axis. Each ordered pair (z_1, z_2) on this plot is a solution to this problem. Observe that both objectives are not arriving at a minimum value on any of the solution patterns. The algorithm nicely generated distinct possible Pareto solutions and as well the alternative solutions within the population count. The total distance is varies between 562 and 866 units where as the total time varies between 436 and 700 units. Thus the decision maker picks an appropriate solution within the acceptable trade-off between the two objectives. Here, the minimum total distance is found to be 562 units. There are 6 distinct route plans which correspond to this minimum total distance with varied total minimum time value in different directions of traversal, listed in Table 5. In Table 5, the first column gives the index of the route plans, the second column contains the direction of the 3-routes for 3-salesmen within each combination of solution, the third and fourth columns respectively give the distance travelled and time taken to cover a subset of cities along the route of a salesman and the last column gives the total time taken by the three salesmen in each combination of a solution. The minimum total

Table 3
Time matrix (t_{ij}).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	17	38	48	19	22	26	39	42	40	5	8	28	1	8	41	13	2	39	7	13	20
2	12	0	4	42	30	15	50	49	36	35	44	8	5	25	49	11	41	44	31	21	29	47
3	17	8	0	13	37	47	20	42	29	1	23	48	47	9	27	12	8	1	41	3	41	20
4	33	32	49	0	49	40	19	18	6	44	42	21	23	33	48	19	42	33	40	39	25	22
5	38	22	22	26	0	38	36	47	25	42	11	38	9	14	50	35	24	29	32	26	46	14
6	29	39	13	9	34	0	7	7	8	13	44	44	1	43	26	34	47	14	4	5	9	39
7	29	34	1	48	9	45	0	38	35	30	4	33	4	44	9	37	44	4	47	28	26	35
8	10	5	24	33	3	1	36	0	16	40	42	21	35	4	36	20	24	8	37	3	30	9
9	8	44	11	17	13	35	33	42	0	4	22	23	43	39	28	9	50	24	37	19	4	33
10	22	3	39	33	4	32	41	30	36	0	49	12	30	16	16	9	27	27	6	48	22	14
11	13	45	12	7	17	32	29	7	16	34	0	21	5	38	14	45	38	26	6	39	2	38
12	2	9	49	9	23	17	3	46	30	3	26	0	45	36	40	50	2	42	2	4	5	1
13	43	9	29	46	33	36	14	23	35	26	5	10	0	33	49	31	5	30	38	32	10	28
14	2	35	41	3	14	22	11	17	43	14	49	49	38	0	44	18	14	34	19	43	30	50
15	3	9	29	1	8	47	25	42	24	13	30	49	34	33	0	32	45	20	44	14	34	33
16	41	8	32	31	7	50	11	17	48	42	27	31	50	8	5	0	28	7	41	25	31	33
17	12	50	5	37	26	28	11	10	10	5	32	7	3	4	15	1	0	28	44	2	10	38
18	45	10	16	29	14	26	44	7	7	7	20	32	6	17	6	21	21	0	36	22	43	31
19	3	28	19	19	39	49	30	13	17	35	37	28	42	34	43	5	43	36	0	7	45	1
20	14	40	26	46	24	40	11	2	26	35	44	43	29	47	31	10	6	23	26	0	6	25
21	25	11	1	49	4	6	24	40	8	26	23	47	34	37	31	9	39	19	10	27	0	36
22	1	32	41	30	14	49	9	16	38	10	33	46	10	28	7	41	14	26	23	26	44	0



a: A feasible solution to BMTSP

b: A set of Pareto solutions of BMTSP with z_1 as first priority

c: Best Pareto solution of BMTSP with z_1 as first priority

d: A set of Pareto solutions of BMTSP with z_2 as first priority

e Best Pareto solution of BMTSP with z_2 as first priority

f Best Pareto solution of BMTSP with 5-salesman

Fig. 6. a: A feasible solution to BMTSP. b: A set of Pareto solutions of BMTSP with z_1 as first priority. c: Best Pareto solution of BMTSP with z_1 as first priority. d: A set of Pareto solutions of BMTSP with z_2 as first priority. e Best Pareto solution of BMTSP with z_2 as first priority. f Best Pareto solution of BMTSP with 5-salesman.

Table 4
A feasible trip schedule for the 3-salesmen.

	Route plan	Distance	Time
Salesman 1	1→10→12→11→3→19→14→15→1	173	212
Salesman 2	1→4→22→6→8→20→17→16→1	179	177
Salesman 3	1→2→7→13→5→18→21→9→1	220	192
Total		572	581

Table 5
The different combination of solutions correspond to the minimum total distance 562 units.

Index	Route plan	Distance covered by a salesman	Time taken by a salesman	Total time
1	1→11→12→10→16→17→8→20→1	204	93	
	1→2→5→13→7→6→22→4→1	178	217	533
	1→15→3→19→14→9→21→18→1	180	223	
2	1→4→22→6→7→13→5→2→1	178	197	
	1→15→3→19→14→9→21→18→1	180	223	547
	1→20→8→17→16→10→12→11→1	204	127	
3	1→18→21→9→14→19→3→15→1	180	160	
	1→2→5→13→7→6→22→4→1	178	217	504
	1→20→8→17→16→10→12→11→1	204	127	
4	1→20→8→17→16→10→12→11→1	204	127	
	1→4→22→6→7→13→5→2→1	178	197	484
	1→18→21→9→14→19→3→15→1	180	160	
5	1→11→12→10→16→17→8→20→1	204	93	
	1→4→22→6→7→13→5→2→1	178	197	450
	1→18→21→9→14→19→3→15→1	180	160	
6	1→18→21→9→14→19→3→15→1	180	160	
	1→2→5→13→7→6→22→4→1	178	217	470
	1→11→12→10→16→17→8→20→1	204	93	

time corresponds to these 6 distinct route plans is observed as 450 units of time. Note that, all the six solutions give the identical total distance because of symmetric distance, whereas these solutions are varied with total time because of asymmetric time matrix. Therefore, the solution that corresponds to index 5 of Table 5 gives the best Pareto solution.

Fig. 6c illustrates the trip schedule of the best found Pareto solution of the BMTSP with z_1 as first priority. The coordinate axes in Fig. 6c assume the locational coordinates. The 22-cities are scattered across the plane and indexed as $1, 2, \ldots, 22$. The three route plans for the three salesman are given by: route 1: 1→11→12→10→16→17→8→20→1, route 2: 1→4→ 22→6→ 7→13→5→2→1 and route 3: 1→18→21→9→14→19→3→15→1. Note that, each salesman starts at depot city 1, traverses through another 7 cities and returns to the depot. The length of each route is $|S_k| = 8$, thus these routes balance the load of the distribution of cities to be visited by a salesman. A salesman can pick up any one of the routes at random to cover the subset of cites. Therefore, the best found solution pair is $(z_1, z_2) = (562, 450)$, in which the first salesman covers 204 units of distance in 93 units of time along route 1, second salesman travel 178 units of distance in 197 units of time through the second route and the third salesman takes route 3, travel 180 units of distance in 160 units of time.

If the salesmen wish to visit all cities with minimum time rather than how much distance they travelled, then in such a scenario set the second objective z_2 as first priority and relax the first objective z_1. Here, there is change in hierarchy of the objective functions only while all other parametric values are unchanged. The GATS algorithm once again operated and obtained a set of ordered Pareto solutions within the threshold population counts by setting z_2 as first priority. The ordered solutions are shown in Fig. 6d. In Fig. 6d, the time parameter taken on the horizontal axis and the distance parameter is on vertical axis. Each of the point is an ordered solution (z_2, z_1) to the BMTSP. Here, the total time is varying between 103 and 630 units and the total distance is varying between 1212 and 1511 units. The lowest value of total time among all the solution pairs is observed as 103 units. Thus, the solution pair $(z_2, z_1) = (103, 1359)$ will be the best Pareto solution when z_2 assumed to be the first priority. The trip schedule of the salesmen corresponds to this solution pair is shown in Fig. 6e and the direction of the route plans for the salesmen are given by route 1: 1→18→10→2→16→15→4→9→1, route 2: 1→11→21→5→13→17→12→22→1 and route 3: 1→14→7→3→20→8→6→19→1. Along route 1, the salesman 1 covered 452 units of distance in 43 units of time, the second salesman covered 412 units of distance in 34 units of time through the second route and the other salesman takes 26 units of time to cover 495 units of distance.

If five salesmen positioned at the depot city instead of three salesmen to cover all the remaining cities, then each salesman has to visit at most $q = \left[\frac{21}{5}\right] = 5$ cities excluding the depot city. Fig. 6 f gives the best route plans for 5-salesmen to cover all the 22 cities. Observe that only one route contains 5 cities and all other routes include only 4 cities other than the common depot. The tour length of a salesman on each route is $|S_k| \leq 6$, which also preserves the load balance constraint. The best Pareto solution of BMTSP with 5-salesman and assuming z_1 as first priority is $(z_1, z_2) = (692, 406)$.

5. Experimental results

This section provides detailed comparative experimental results on some benchmark instances created for MTSP, the standard collection available in TSPLIB. To evaluate the capabilities of the proposed GATS, the algorithm was nicely coded in MATLAB-2019b and executed on Microsoft Windows11 version 22H2, operating system with an Intel(R) Core (TM) 8250 U processor running at 1.60 GHz or 1.80 GHz and 8 GB of memory. Firstly, the proposed BMTSP is reduced to an MTSP with load balance constraint by ignoring the second objective function z_2 alone in the model to compare the GATS results with the results reported by Carter and Ragsdale [15], Brown et al. [12] and Alok and Anurag [7]. For computational experiments, we selected the same test instances used in Carter and Ragsdale [15] and Brown et al. [12]. These test instances are of sizes 51, 100 and 150 cities and also available in TSPLIB as Eil51, KroA100, and KroB150 respectively. These test instances consists of two-dimensional Euclidean locational coordinates of cities. The distance between a pair of cities is generated using the usual geometric distance formula. Thus, the distance between a pair of cites is observed to be symmetric. In each test instance, there are three test cases with respect to each value of $m = \{3, 5, 10\}$. So there are 9 test problems altogether used for comparison. All the experiments carried out with GATS have used the population size of 100 chromosomes. Each salesman has to visit a maximum of $q = \left[\frac{n-1}{m}\right]$ cities to balance the load on visiting the cities. Performed ten repeated runs on each test problem with the proposed GATS, each time a different random chromosome is used.

Carter and Ragsdale [15] proposed different combinations of GA's with one chromosome (GA1C), two chromosome (GA2C), and two-part chromosome (GA2PC), Brown et al. [12] used GA1C, GA2C and in addition a grouping genetic algorithm (GGA), and Alok and Anurag [7] developed a steady-state grouping genetic algorithm (GGA-SS), recorded the solutions on MTSP with restrictions on the solutions as the tour of each salesman starts and ends at the depot city, and a salesman must have to visit at least one city other than the depot. If the value of m increases, the overall distance travelled by all the m-salesman will also increase. In addition to these constraints, we also added a load balance constraint as well to the proposed GATS algorithm, where each salesman has to visit no more than q cities, which sets an upper limit on the number of cities. Adding this load balance constraint to the MTSP, the total distance travelled by all the salesmen will also increase than the usual MTSP due to the fact that a salesman cannot visit more than q cities.

The comparative performance of the proposed GATS is shown in Table 6. It is observed that GATS provides better solutions than the GA1C, GA2C proposed by Carter and Ragsdale [15] on all the test

Table 6
Performance of GATS verses [12,15] and [7].

Instance	m	Carter and Ragsdale[15] GA1C	GA2C	GA2PC	Brown et al.[12] GA1C	GA2C	GGA	Alok and Anurag [7] GGA-SS	Proposed GATS	Time (s)
Eil51	3	529	570	543	596	951	924	449	467	18.78
	5	564	627	586	726	1180	882	479	553	18.37
	10	801	879	723	1015	1403	1001	584	779	20.92
KroA100	3	27036	30972	26653	46,929	80,195	79,347	22051	24823	17.32
	5	29753	44062	30408	51,509	97,741	70,871	23678	28345	18.57
	10	36890	65116	31227	68,664	129,000	89,778	28488	42468	23.10
KroB150	3	46111	48108	47418	17,754	37,373	33,888	38434	32375	19.02
	5	49443	51101	49947	20,558	38,766	26,851	39962	39996	33.58
	10	59341	64893	54958	27,621	39,906	37,771	44274	55595	42.61

instances considered. However, when it is compared with GA2PC, except for $m = 10$, remaining test cases the GATS produced the most prominent solutions. On the instances Eil51 and KroA100, the GATS showed superiority for all distinct values of m, than that of GA1C, GA2C and GGA proposed by Brown et al. [12], while on the instance KroB150, GATS obtained improved solutions than GA2C and GGA for $m = 3$ only and for other values of m, the results of GATS are optimistic when compared with other algorithms. Although the results of GATS are differing with results of GGA-SS proposed by Alok and Anurag [7] for distinct values of m, the difference would not be statistically significant at 0.01 level of significance. The GATS provide an approximate solution rather an exact solution and as well it is taking more time while solving higher dimension instances.

For better understanding, the comparative results reported in Table 6 are shown in appropriate bar plots from Figs. 7a-7g. Figs. 7a-7c show the comparative bar plots of GATS results with Carter and Ragsdale [15] on the instances with cities 51, 100 and 150 for distinct values of m. Observe that, the GATS is obtained better solutions when $m = 3$ and $m = 5$, while m= 10 obtained a mixed results. Figs. 7d-7f show the comparative bar plots of GATS results with Brown et al. [12] on the instances with cities 51, 100 and 150 for distinct values of m. Observe that, the GATS obtained high quality solutions on the instances Eil51 and KroA100 for all considered values of m, while on the instance KroB150 GATS obtained mixed results on different values of m. Fig. 7 g show that the results obtained by GGA-SS and GATS have good agreement on the instances experimented.

Further, the comparative study is extended on the recent works appeared on the MTSP. Table 7 present the comparative results between the GATS and the memetic algorithm suggested by AI- Taani and AI-Afifi, 2022. The experiments are carried on six distinct benchmark instances from TSPLIB. On each instance we considered 5 test cases by varying the parametric value $m = 2, 3, 4, 5 and 6$ and resulting a total of 30 test cases. Among the 30 test cases the GATS obtained improved solutions in 20 test cases. Observe that, the GATS generated high quality solutions than the memetic algorithm for all considered values of m on the instances Pr124 and Pr152. The average of all the solutions obtained by memetic algorithm is 101306.2 and the average of all the solutions obtained by GATS is 70398.62. The deviation on the average solution value on all the considered test cases is approximately 30%. However, the t-test says that both the results are not statistically significant at 0.05 and 0.01 levels of significance.

Table 8, describes the comparison of the proposed GATS results with the results reported by Xiaolong Xu et al. (2018) on different benchmark TSP instances. In this table, optimal Hamiltonian circuit (OHC) gives the best known solution on the test instances considered, whereas the other algorithms such as ant colony algorithm (ACA) proposed by Liu et al. [38], the nearest neighbour method (NNM) by Wang [57] and an improved GA by Xiaolong Xu et al. (2018) obtained the nearer solutions. These algorithms are designed to find least distance travelled by a single salesman to cover all the cities. For example, on the instance Eil51, the salesman will travel 426 units of distance to cover all 51 cities with OHC, while 435 units with ACA, 453 units with NNM, 443 units with improved GA and 428 units with the proposed GATS. The deviation between the best found solution and the best known solution is called error and the percentage error is computed using the Eq. (11).

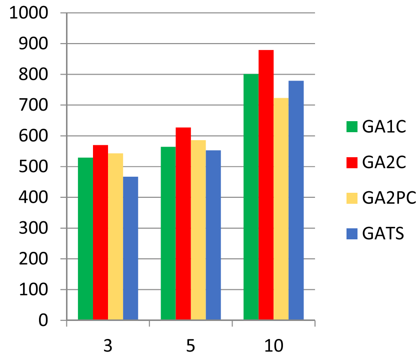
$$err = \frac{w(Best) - w(OHC)}{w(OHC)} \times 100 \tag{11}$$

Note that a small error rate indicates that the algorithms used have achieved better solutions. Fig. 8 illustrates the comparison of GATS with ACA, NNM and improved GA algorithms on error rate. Because GATS has a lower error rate, it is capable of producing prominent solutions.

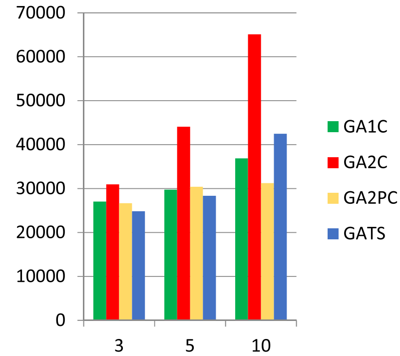
In addition, Xiaolong Xu et al. (2018) selected 16 scenic locations in Nanjing city (China), which a tourist wants to visit in $m = 4$ days from a hotel (depot) and each day not more than $q = 4$ locations such that the overall distance travelled by the tourist is minimum. The distance between the locations is calculated with the help of the geographical coordinates, then a two-phase heuristic algorithm (TPHA) is used to find the best routes for the tourist. The scatter plot of coordinates of the 16 scenic locations and a depot point of tourist are shown in Fig. 9a. Further, the proposed GATS was tested on this instance as well and its performance with TPHA and GA is shown in Table 9. It is observed that there is an ignorable deviation between the GATS and TPHA solutions. This may be due to the change in the coordinates of the location NGG as its coordinates are not available rightly in their paper. However, the route plans obtained with GATS for the tourist, shown in Fig. 9b, remains same as TPHA route plans. The GATS is taking more time than the two algorithms TPHA and GA, but the solution obtained by GATS is significantly better than GA.

6. Conclusions and future works

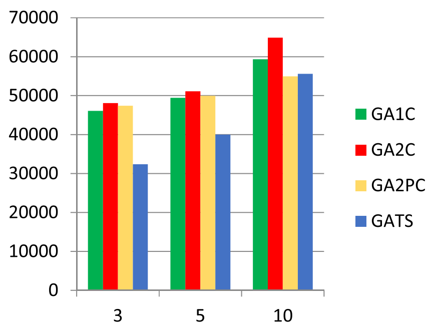
This paper addresses a bi-objective version of MTSP with a load balance constraint. As MTSP is NP-hard, its extended version, BMTSP will also be an NP-hard nature problem. BMTSP is formulated with the help of zero-one linear programming. The study contributes a metaheuristic-based genetic algorithm with tournament selection that integrates distinct mutation strategies to solve the BMTSP. The different mutation strategies used in GATS helped to resolve the local minima problems of premature convergence to attain better solutions. A detailed discussion of the BMTSP and the methodology of GATS is provided through an appropriate numerical illustration. Further, GATS experimented on the various benchmark instances, and the results reveal that GATS obtained prominent solutions on a wide range of benchmark instances and was quite competitive when compared with different versions of the existing GA's considered on the MTSP. Also, the proposed GATS produced high-quality solutions on 20 test cases out of 30 test cases when compared with the memtic algorithm. Adding to this, the result of GATS outperforms the TSP case when compared with the other existing algorithms, such as ACA, NNM and an improved GA. Further, the solutions obtained with GATS and TPHA have good agreement. The GATS provide an approximate solution rather than an exact solution, and it is taking more time as the size of the instance and the number of salesmen increase.



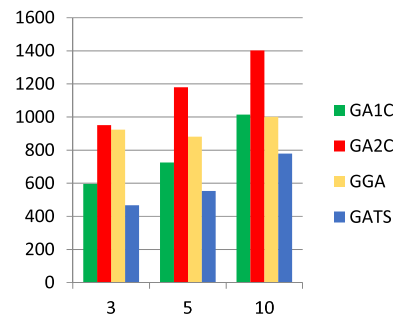
a: Comparison of GATS with Carter and Ragsdale (2006) on the instance Eil51



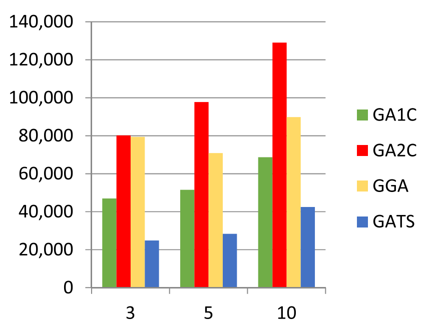
b: Comparison of GATS with Carter and Ragsdale (2006) on the instance KroA100



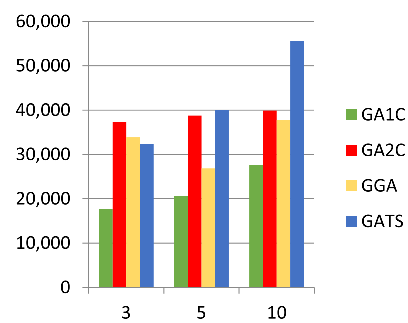
c: Comparison of GATS with Carter and Ragsdale (2006) on the instance KroB150



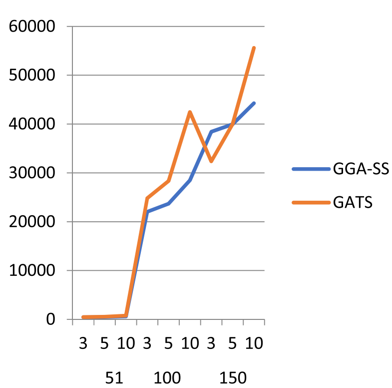
d: Comparison of GATS with Brown et al. (2007) on the instance Eil51



e: Comparison of GATS with Brown et al. (2007) on the instance KroA100



f: Comparison of GATS with Brown et al. (2007) on the instance KroB150



g: Comparison of GATS with Alok and Anurag (2009) on the instance KroB150

Fig. 7. a: Comparison of GATS with Carter and Ragsdale [15] on the instance Eil51. b: Comparison of GATS with Carter and Ragsdale [15] on the instance KroA100. c: Comparison of GATS with Carter and Ragsdale [15] on the instance KroB150. d: Comparison of GATS with Brown et al. [12] on the instance Eil51. e: Comparison of GATS with Brown et al. [12] on the instance KroA100. f: Comparison of GATS with Brown et al. [12] on the instance KroB150. g: Comparison of GATS with Alok and Anurag [7] on the instance KroB150.

Table 7
Comparison between GATS and Memetic algorithm (AI- Taani and AI-Afifi, 2022).

Instance	#m	Memetic algorithm using GA and HC	Proposed (GATS)	Instance	#m	Memetic algorithm using GA and HC	Proposed (GATS)
Eil51	2	467	**446.67**	Rat99	2	5365	**1460.25**
	3	600	**467**		3	1839	**1666.99**
	4	499	509		4	1927	**1841.19**
	5	503	553		5	1970	2080.01
	6	501	580.74		6	2025	2244.54
Att48	2	38318	**34761.29**	Pr124	2	170954	**71800.18**
	3	38647	**38268.32**		3	168439	**77780.83**
	4	38880	44682.48		4	172917	**95010.14**
	5	39634	49300.56		5	175733	**102543**
	6	40943	55322.90		6	174731	**111877.57**
Pr76	2	150615	**123188.88**	Pr152	2	189281	**91984.33**
	3	154047	**135300.53**		3	397530	**112050.08**
	4	162474	**155026.46**		4	225065	**121057.99**
	5	162984	171401		5	224794	**148055**
	6	173295	190119.79		6	224210	**170578**

Table 8
Comparison of GATS with the results reported by Xiaolong Xu et al. (2018).

TSP Instances	OHC	ACA		NNM		The improved GA		GATS	
		Best	err	Best	err	Best	err	Best	err
Eil51	426	435	2.11	453	6.34	443	3.74	428	0.47
Eil76	538	551	2.41	582	8.18	568	5.57	551	2.41
Eil101	629	682	8.42	715	13.6	693	9.65	655	4.13
Berlin52	7542	7543	0.01	7976	5.57	7644	1.32	7542	0

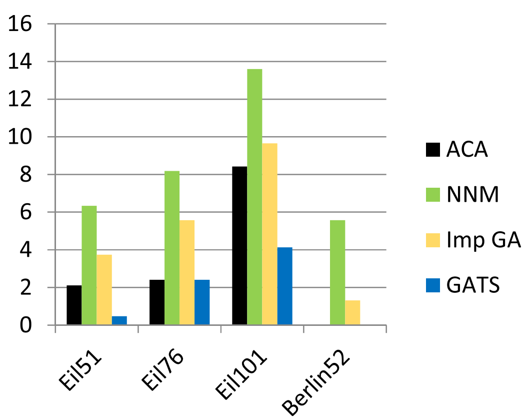
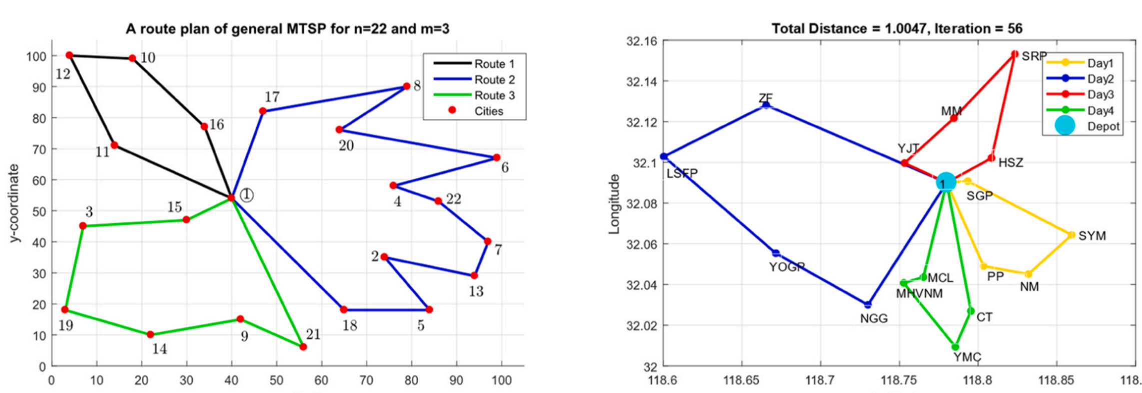


Fig. 8. Comparison of GATS with Xiaolong (2018) on the error term, given in Table 8.

In the future, the proposed BMTSP may be extended to formulate multiple objectives with multiple depots and impose additional constraints that can accommodate more of the real-world problems that appear in humanitarian logistics, supply chain management, personal rapid transit systems, service management, managerial decision-making, etc. In addition, we are planning to investigate the use of a new combination of crossover operators and embed them with distinct local search algorithms and other evolutionary optimization algorithms to generate more accurate solutions.

Table 9
Performance of GATS with TPHA and GA.

Algorithm	Distance	Time (sec.)
TPHA	0.9941	2.442360
GA	1.1648	10.698919
GATS	1.0047	13.968180



a The scatter plot of coordinates of scenic locations with depot

b The best route plan for the tourist with GATS

Fig. 9. a The scatter plot of coordinates of scenic locations with depot. b The best route plan for the tourist with GATS.

Declaration of Competing Interest

Authors express that no conflicts of interests.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful comments to help us improve the quality of this work.

References

[1] B. Abdollahzadeh, G.F. Soleimanian, S. Barshandeh, Discrete farmland fertility optimization algorithm with metropolis acceptance criterion for traveling salesman problems, Int. J. Intell. Syst. 3 (2020) 1270–1303.
[2] N. Ahmed, S. Das, S. Purusotham, The oil tankers dispatching problem, OPSEARCH 49 (2012) 366–385.
[3] Z.H. Ahmed, Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator, Int. J. Biom. Bioinforma. (IJBB) 3 (6) (2010) 96.
[4] A.T. Al-Taani, L.M. Al-Afifi, Solving the multiple traveling salesman problem using memetic algorithm, Artif. Intell. Evol. 17 (2022) 27–40.
[5] H. Ali, H.D. Ahmed, R. Omar, A new local search for the bees algorithm to optimize multiple traveling salesman problem, Intell. Syst. Appl. 18 (2023) 200242.
[6] A.I. Ali, J.L. Kennington, The asymmetric M-travelling salesmen problem: a duality-based branch-and- bound algorithm, Discret. Appl. Math. 13 (1986) 259–276.
[7] Alok Singh, A.S. Baghel, A new grouping genetic algorithm approach to the multiple traveling salesperson problem, Soft Comput. 13 (2009) 95–101.
[8] R.M. Alves, C.R. Lopes, Using genetic algorithms to minimize the distance and balance the routes for the multiple traveling salesman problem, IEEE Congr. Evolut. Comput. (CEC) (2015) 3171–3178.
[9] K. Andras, J. Abonyi, Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm. intelligent computational optimization in engineering, Stud. Comput. Intell. 366 (2011).
[10] R.D. Angel, W.L. Caudle, R. Noonan, A. Whinston, Computer assisted school bus scheduling, Manag. Sci. 18 (1972) 279–288.
[11] V. Bhavani, M. Sundara Murthy, Truncated M-travelling salesmen problem, Opsearch 43 (2) (2006) 152–177, 2006.
[12] E. Brown, C.T. Ragsdale, A.E. Carter, A grouping genetic algorithm for the multiple traveling salesperson problem, Int. J. Inf. Technol. Decis. Mak. (IJITDM) 6 (2) (2007) 333–347.
[13] R.W. Calvo, R. Cordone, A heuristic approach to the overnight security service problem, Comput. Oper. Res. 30 (9) (2003) 1269–1287.
[14] A.E. Carter, C.T. Ragsdale, Scheduling pre-printed newspaper advertising inserts using genetic algorithms, Omega 30 (6) (2002) 415–421.
[15] A.E. Carter, C.T. Ragsdale, A new approach to solving the multiple travelling salesperson problems using genetic algorithms. Eur. J. Oper. Res. 175 (1) (2006) 246–257.
[16] Chao Jiang, Zhongping Wan, Zhenhua Peng, A new efficient hybrid algorithm for large scale multiple traveling salesman problems, Expert Syst. Appl. 139 (2020) 112867.
[17] Shih-Hsin Chen, Mu-Chung Chen, Operators of the two-part encoding genetic algorithm in solving the multiple traveling salesmen problem, Conf. Technol. Appl. Artif. Intell. (TAAI (2011).
[18] L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.
[19] R. Dhanalakshmi, P. Parthiban, N. Anbuchezhian, Optimisation of multiple travelling salesman problem using metaheuristic methods, Int. J. Enterp. Netw. Manag. 3 (3) (2022) 199–215.
[20] S.G. Farhad, A. Benyamin, An efficient harris hawk optimization algorithm for solving the travelling salesman problem, Clust. Comput. 25 (3) (2021) 1981–2005.
[21] S.G. Farhad, A. Benyamin, A. Bahman, An improved farmland fertility algorithm with hyper-heuristic approach for solving travelling salesman problem, Comput. Model. Eng. Sci. 135 (3) (2023) 1981–2006.
[22] O.I. Farisi, B. Setiyono, R.I. Danandjojo, A hybrid approach to multi-depot multiple traveling salesman problem based on firefly algorithm and ant colony optimization, IAES Int. J. Artif. Intell. 10 (4) (2021) 910.
[23] B. Gavish, K. Srikanth, An optimal solution method for large-scale multiple traveling salesmen problems, Oper. Res. 34 (1986) 698–717.
[24] G. Giardini, T. Kalmar-Nagy, Genetic algorithm for multi-agent space exploration, Proc. AIAA Info-Tech. Aerosp. Conf. 2 (2007) 1146–1160.
[25] D.E. Goldberg, Genetic algorithm in search. Optimization and Learning, Addison-Wesley, Reading, MA, 1989.
[26] B. Hamdan, H. Bashir, A. Cheaitou, A novel clustering method for breaking down the symmetric multiple traveling salesman problem, J. Ind. Eng. Manag. 14 (2) (2021) 199–218.
[27] P. He, J.K. Hao, Memetic search for the minmax multiple traveling salesman problem with single and multiple depots, Eur. J. Oper. Res. 307 (3) (2023) 1055–1070.
[28] J.H. Holland. Adaptation in natural and artificial systems, first ed., MIT Press, Cambridge, MA, 1975, 2nd ed., 1992.
[29] T. Jayanth Kumar, S. Purusotham, An open-close multiple travelling salesman problems with a single Depot, Decis. Sci. Lett. 8 (2019) 121–136.
[30] T. Jayanth Kumar, S. Purusotham, Designing an optimal route for the distribution chain of a rural LPG delivery system, Int. J. Ind. Eng. Comput. (2021) 221–234.
[31] Joris K., Bart S., Eline D.F. and Spieksma, C.R. Exact algorithms for the Equitable Traveling Salesman Problem. 261(2), 475–485, (2017).
[32] P. Junjie, W. Dingwei, An ant colony optimization algorithm for multiple travelling salesman problem, Innov. Comput., Inf. Control, IEEE 1 (2006) 210–213.
[33] K. Kloster, M. Moeini, D. Vigo, O. Wendt, The multiple traveling salesman problem in presence of drone-and robot-supported packet stations, Eur. J. Oper. Res. 305 (2) (2023) 630–643.
[34] G. Laporte, A concise guide to the TSP, J. Oper. Res. Soc. 61 (2010) 35–40.
[35] G. Laporte, Y. Nobert, A cutting planes algorithm for the m-salesmen problem, J. Oper. Res. Soc. (1980) 1017–1023.
[36] Z. Ling, Y. Zhou, Y. Zhang, Solving multiple travelling salesman problem through deep convolutional neural network, IET Cyber Robot. 5 (1) (2023) e12084.
[37] M. Liu, P.Y. Zhang, New hybrid genetic algorithm for solving the multiple traveling salesman problem: an example of distribution of emergence materials, J. Syst. Manag. 23 (02) (2014) 247–254.
[38] Y. Liu, X. Shen, H. Chen, An adaptive ant colony algorithm based on common information for solving the traveling salesman problem. Proceedings Shandong, ICSAI,, 2012, pp. 763–766.
[39] K.M. Lo, W.Y. Yi, P.K. Wong, K.S. Leung, Y. Leung, S.T. Mak, A genetic algorithm with new local operators for multiple traveling salesman problems, Int. J. Comput. Intell. Syst. 11 (1) (2018) 692–705.
[40] B. Maria, P. Artur, S. Anand, U. Eduardo, Exact algorithms for the traveling salesman problem with draft limits, Eur. J. Oper. Res. 235 (1) (2014) 115–128.
[41] B.L. Miller, D.E. Goldberg, Genetic algorithms, tournament selection, and the effects of noise, Complex Syst. 9 (3) (1995) 193–212.
[42] C. Okonjo-Adigwe, An effective method of balancing the workload amongst salesmen, Omega 16 (2) (1988) 159–163.
[43] S. Purusotham, T. Jayanth Kumar, An efficient genetic algorithm for solving open multiple travelling salesman problem with load balancing constraint, Decis. Sci. Lett. 10 (4) (2021) 525–534.
[44] B. Ruben Ivan, M.T.O. Eliana, G.E. Mauricio, A population-based algorithm for the multi travelling salesman problem, Int. J. Ind. Eng. Comput. 7 (2016) 245–256.
[45] R.A. Russell, An effective heuristic for the m-tour travelling salesman problem with some side conditions, Oper. Res. 25 (3) (1977) 517–524.
[46] M. Sedighpour, M. Yousefikhoshbakht, N. Mahmoodi Darani, An effective genetic algorithm for solving the multiple traveling salesman problem, J. Optim. Ind. Eng. 8 (2011) 73–79.
[47] Y. Shuai, S. Yunfeng, Z. Kai, An effective method for solving multiple travelling salesman problem based on NSGA-II, Syst. Sci. Control Eng. 7 (2019) 121–129.
[48] S. Sivanandam, S. Deepa, Genetic algorithm optimization problems. Introduction to Genetic Algorithms, Springer, 2007.
[49] M. Snezana, K. Ramesh, The multiple TSP with time windows: Vehicle bounds based on precedence graphs, Oper. Res. Lett. 34 (1) (2006) 111–120.
[50] C. Song, K. Lee, W.D. Lee, Extended simulated annealing for augmented TSP and multi-salesmen TSP, Proc. Int. Jt. Conf. Neural Netw. 3 (2003) 2340–2343.
[51] Sze, S., & Tiong, W. A comparison between heuristic and meta-heuristic methods for solving the multiple traveling salesman problem. World Academy of Science, Engineering and Technology, (2007).
[52] K.N. Tamás, G. Giovanni, The multi-agent planning problem, Proc. 3rd Int. Conf. Agents Artif. Intell. (ICAART) 2 (2011) 296–305.
[53] J. Tang, A. Liu, Z. Rong, A. Yang, multiple travelling salesman problem model for hot rolling scheduling in Shangai Baoshan iron and steel complex, Eur. J. Oper. Res. 124 (2000) 1267–1282.
[54] Toth, P. and Vigo, D. The Vehicle routing problem, Philadelpia: Society for Industrial and Applied Mathematics, Series: SIAM monographs on discrete mathematics and applications, (2002).
[55] Venkatesh Pandiri, Alok Singh, A hyper-heuristic based artificial bee colony algorithm for k-interconnected multi-depot multi-traveling salesman problem, 261-81, Inf. Sci. 463 (2018), 261-81.
[56] P. Venkatesh, Alok Singh, Two metaheuristic approaches for the multiple traveling salesperson problem, Appl. Soft Comput. 26 (2015) 74–89.
[57] Y. Wang, A nearest neighbor method with a frequency graph for traveling salesman problem, in: Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics, 1, IEEE, 2014, pp. 335–338.
[58] X. Xiaolong, H. Yuan, M. Liptrott, M. Trovati, Two phase heuristic algorithm for the multiple-travelling salesman problem, 6567-81, Soft Comput. 22 (2018), 6567-81.
[59] M. Yousefikhoshbakh, M. Seidighpour, A combination of sweep algorithm and elite ant colony optimization for solving the multiple traveling salesman problem, Proc. Rom. Acad. A 13 (4) (2012) 295–301.
[60] Youssef Harrath, Three-step metaheuristic for the multiple objective multiple traveling salesman problem, Int. J. Appl. Metaheur. Comput. IGI Glob. 11 (4) (2020) 130–148.
[61] H. Youssef, A.F. Salman, A. Alqaddoumi, H. Hasan, A. Radhi, A novel hybrid approach for solving the multiple traveling salesmen problem, 103-12, Arab J. Basic Appl. Sci. 26 (1) (2019), 103-12.
[62] S. Yuan, B. Skinner, S. Huang, D. Liu, A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms, Eur. J. Oper. Res. 228 (1) (2013) 72–82.

[63] J. Zheng, Y. Hong, W. Xu, W. Li, Y. Chen, An effective iterated two-stage heuristic algorithm for the multiple Traveling Salesmen Problem, Comput. Oper. Res. 143 (2022) 105772.
[64] H. Zhou, M. Song, W. Pedrycz, A comparative study of improved GA and PSO in solving multiple traveling salesmen problem, 564-80, Appl. Soft Comput. 64 (2018), 564-80.
[65] W. Zhou, Y. Li, Improved genetic algorithm for multiple traveling salesman problem. Informatics in Control, Automation and Robotics (CAR), 2010 2nd Int. Asia Conf. 1 (2010) 493–495.