Alexandria Engineering Journal (2023) 64, 349–363

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows

Zakir Hussain Ahmed [a,*], Majid Yousefikhoshbakht [b]

[a] Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Kingdom of Saudi Arabia
[b] Department of Mathematics, Faculty of Sciences, Bu-Ali Sina University, Hamedan, Iran

Received 10 May 2022; revised 26 July 2022; accepted 3 September 2022
Available online 14 September 2022

KEYWORDS

Open Vehicle Routing Problem with Time Window;
Fixed fleet Heterogeneous;
Tabu Search Algorithm;
Mixed Integer Linear Programming

Abstract The heterogeneous fixed fleet open vehicle routing problem with time windows is a very significant type of the vehicle routing problem (VRP) that aims to find the minimum fixed and variable cost of transportation for a heterogeneous fleet with a fixed number in which the capacity of every vehicle and usage of the vehicles should not be ignored. Also, in this problem, each customer has a special time window for servicing and each vehicle starts its route from the warehouse and ends up in one of the customers. We propose a mixed integer linear programming model of this problem. Since this problem, as well as open VRP and VRP with fixed heterogeneous fleet are hard NP problem, an improved tabu search algorithm is proposed to solve the problem. Our proposed algorithm uses a modified sweep algorithm to generate some initial solutions. Besides, a variable tabu list and some new mechanisms for intensification and diversification mechanisms are used. Numerical results are presented to show the correctness of our model and finally, the efficiency of the proposed algorithm is compared with an exact algorithm, classic tabu search and simulated annealing. The obtained results prove the efficiency of the proposed algorithm.

1. Introduction

One of the ways to reduce the cost of producing a product is to minimize the cost of transportation so that it can be transferred from place to place with the least cost. Therefore, nowadays, the importance of the vehicle routing problem (VRP) and its spreads is not hidden to anyone, and its real applications in daily life have led researchers to give more importance to it day-by-day and examine it from different angles [1,2]. On the other side, this problem has very significant character in the distribution management and supply chain so that over the last 50 years, many extensions, such as open VRP [3], VRP with simultaneous pickup and delivery [4], dynamic VRP [5], VRP with time windows [6], fixed heterogeneous

* Corresponding author.
E-mail addresses: zaahmed@imamu.edu.sa (Z. Hussain Ahmed), khoshbakht@basu.ac.ir (M. Yousefikhoshbakht).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2022.09.008
1110-0168

VRP [7] and distance-constrained VRP [8] are presented. In the VRP, a fleet of vehicles in the warehouse must serve a number of customers scattered around the warehouse provided that the total demand of the customers allocated to each vehicle does not exceed the fixed capacity, Q, and each customer is served by only one of these vehicles. In this problem, in addition to presuming that all vehicles are alike, the vehicles must deliver only one kind of items to the customers.

On the other hand, today, the manufacturing companies do not market, transport and sell their products/items, but leave them to some specialized companies. Therefore, outsourcing and use of specialized companies have become very common today and many investments have been made on transportation contracts for distribution of their products by specialized companies. In these contracts, shipping companies take their fleet to the warehouse of the manufacturer's products and deliver their products to other customers. The vehicles in these cases, so called stationary, will not come back to the warehouse after finishing their jobs, but end their paths in the end customers. This problem is called the open VRP (OVRP), wherein Hamilton routes of every vehicle are open, unlike the VRP that are closed. For example, Fig. 1 shows solutions to the VRP and the related OVRP instances where there are 26 customers and 3 vehicles for the VRP instance (the left figure), and 26 customers and 7 vehicles for the OVRP instance (the right figure). In this figure, it is presumed that for every vehicle, the total demand of the customers is less than the capacity of the vehicle. Also, the horizontal and vertical axes in this figure show the coordinates of the points in the 2D plane. In other words, it is assumed that the customers are distributed on a two-dimensional page.

Postal shipments, distribution of newspapers to homes, etc. are some applications of the OVRP. For more applications, one can see in [9,10]. This is also not a long history, contrary to the VRP, it was first introduced by Sarikilis and Powell in 2000 [11]. For this reason, they presented a first-clustering-second routing method for solving the OVRP. Since then, this problem has been considered by several researchers and many methods have been presented to solve it, some of which are genetic algorithm [12], iterated local search [13], tabu search [14], and variable neighbourhood search [15].

Consideration of open routes due to the rental of vehicles has also been used in other versions of the VRP. For example, if the fleet used in the heterogeneous fixed fleet VRP (HFFVRP) is least, it becomes a heterogeneous fixed fleet OVRP (HFFOVRP) which has important applications in transportation and logistics problems. In this HFFOVRP, a fleet of a fixed number of vehicles with different capacities serve a number of customers with a specific request. The optimal usage of the vehicles is expected to provide services to the customers so that it results in minimum cost and every vehicle obeys the following conditions.

- The vehicles are different in regards of their capacity, fixed cost (hiring cost, maintenance cost) and variable cost (per distance unit cost) and there is a fixed and specific number of every type.
- Each vehicle starts its route from the warehouse and finishes its route in the last visited customer.
- The total amount of request for goods by the customers is less than the capacity of the assigned vehicle.
- The amount of acceptable time for each vehicle is less than the maximum specified time for that vehicle.

The HFFOVRP is a most recent version of the VRP that was first considered in 2012 [16]. Contrary to the OVRP, due to depreciation and lack of proper efficiency, it is better for the companies to use vehicles with different capacities because they may require different goods and may have different economic conditions. Also, in most cases, transportation costs to transport goods from heterogeneous fleet of vehicles can be less costly. In addition, in many applications of this problem, it may be necessary for the goods to arrive for each customer within a specified time period. Especially, in perishable goods where there is little time to deliver goods which are sold for specific hours a day and customers prefer to receive them at a specific time. Therefore, considering time windows for the HFFOVRP leads to a problem called HFFOVRP with time windows (HFFOVRPTW), which has more applications in industry and services. In this study, a mixed integer linear programming (MILP) model for the HFFOVRPTW is suggested and then several instances of the problem are solved by an exact method (AIMMS software version 12.3). Also, this is an expanded problem of the OVRP and the HFFVRP that can be addressed by reducing limitations. For example, if, whatever the time windows, it is supposed that the vehicles

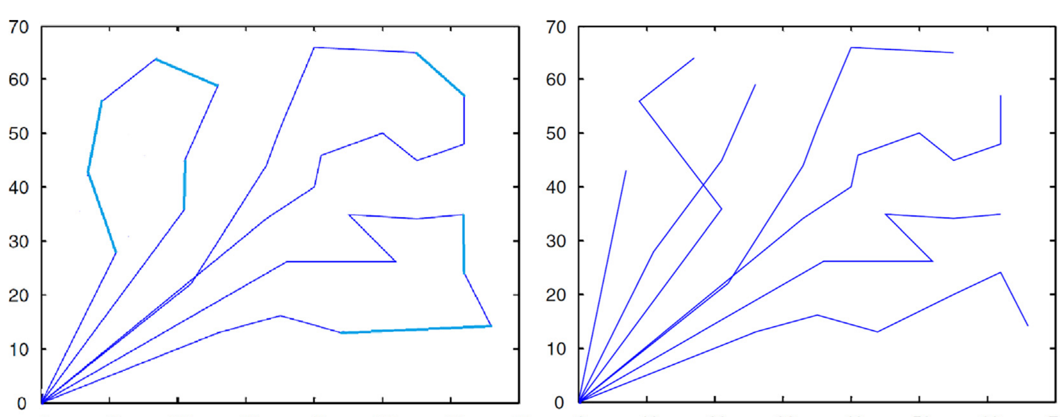

Fig. 1 Solutions to the VRP (the left figure) and the related OVRP (the right figure) instances.

are homogeneous with their same capacity, then the problem turns into the OVRP. Since in the basic OVRP, no fixed and variable cost for using the vehicles is present, so, the fixed cost and the variable cost are supposed to be zero and one respectively. On the other side, if the Hamilton path restriction is changed to Hamilton cycle for the HFFOVRP without time windows, then the HFFVRP will be transferred. This conversion can easily be done by considering zero transportation cost between each end customer and the warehouse. About the complexity, since the OVRP and the HFFVRP are NP- hard, and the HFFOVRPTW is also NP-hard [17], that means, the problem is very difficult to solve using exact methods. So, one can go for the heuristic/metaheuristic algorithms and tabu search (TS) [14] is one of the good heuristic algorithm. Further, many hybrid algorithms combining different heuristic/metaheuristic algorithms are proposed in the literature to solve some constrained optimization problems, like hybrid particle swarm optimization - genetic algorithm [18], hybrid gravitational search algorithm- genetic algorithm [19], hybrid lexisearch-genetic algorithm [20], etc. Hence, we aim to develop an improved TS (ITS) algorithm for solving the HFFOVRPTW. The details aim and objective of this study are as follows.

- To develop a mixed integer linear programming (MILP) model for the HFFOVRPTW.
- To solve several instances of the problem by an exact method (AIMMS software version 12.3).
- To develop ITS algorithm for solving the problem.
- To carry out a comparative study among ITS, an exact method, basic TS and simulated annealing (SA) algorithms on some benchmark problem instances.
- To prove the efficiency of the proposed ITS algorithm.

The paper is arranged as follows: first, the history of the problem is examined in section 2 and then the problem model is presented in section 3. In section 4, the proposed improved TS method is described and then in section 5, several real instances are considered and solved by different algorithms. Finally, conclusions and future directions are presented in section 6.

2. Literature review

In the transportation industry, the homogeneous fleet, in which the vehicles are fully similar, is used to provide different services to customers because a fleet is usually purchased for a long period of time, and the owners of transportation companies typically tend to have a fleet of different vehicles to work in different fields. Also, due to the differences in the type of goods or the type of distribution of customers around manufacturing companies, many companies closed their contracts for the distribution of goods with different service companies, each of which has a similar fleet of vehicles. In other words, there are situations where manufacturing companies use several fleets to distribute their goods with a lower cost. As a results emphasis has been shifted from using homogeneous fleet to the heterogeneous fleet of vehicles by different manufacturing companies. Therefore, heterogeneous fleet VRP (HFVRP) has numerous service and industrial applications and has drawn interest to numerous researchers who developed various algorithms [21–23]. In this kind of fleet problems characteristics of the vehicles may be heterogeneous having different capacities, the number of vehicles in a limited and specified way, the fixed cost for using each vehicle and the variable cost for using any vehicle in the distance unit.

In 2005, Chu [24] examined a new version of the HFVRP, in which a combined fleet is intended to serve customers. The fleet had a number of vehicles that could return to the warehouse after servicing customers because those vehicles belonged to the product manufacturer company those had to be used for later service in the coming days. On the other hand, this fleet usually cannot meet all the company's requirements for delivery of goods all year long and may require a number of more vehicles on some days of the year, in which case a number of other rental vehicles are needed. Because this new version was the HFVRP, a mathematical modeling for the problem was presented and then it was solved by a heuristic method. In this problem, customers who need to be visited by rental vehicles are of great importance and in this reference a new method has been used for this choice. Considering the importance of rental vehicles that can solve the problem if needed more than the capacity of the vehicles of the company, this modification was used again in [25]. In their paper, the researchers utilized multiple local heuristic method to enhance the initial solution and then after obtaining the second solution, they tried to enhance this solution again using the improved method. This was repeated till the algorithm's termination criterion was achieved. The use of different enhancing methods in this method led to have better search space and hence, it was capable to deliver high-quality solutions. Prins [26] examined the HFVRP more closely and presented a hybrid metaheuristic algorithm to solve both versions of the problem, the HFFVRP and the fleet size mix VRP (FSMVRP). The algorithm is based on genetic algorithm and applied local search methods to enhance the effectiveness of the method in the intensification process. The changes done in the phases of the method improved the effectiveness compared to the basic GA version. A real application of the HFFVRP was presented by Schmeda et al. [27] for the distribution of concrete for a construction company in 2010. In this study, which used 20 real instances for experimentation, an efficient algorithm was presented that started based on an exact algorithm. In addition, in order to prevent the high implementation time of the exact algorithm, the obtained solutions were considered by this algorithm and the quality of them was improved using variable large neighborhood search algorithm. The combination of these two algorithms resulted in very good solutions at the acceptable time for instances. In addition, a TS *meta*-efficiency algorithm was developed in 2010 for the HFFVRP by Euchi and Chabchoub [28]. In this algorithm, a comparative domain was used to further improve, and several instances were generated to test its efficiency. Brandao [17] developed a deterministic TS algorithm for the FSMVRP and presented examples. Brandao [29] further developed TS algorithm for solving the HFFVRP. Also in this paper, some problem instances are created, and the effectiveness of the algorithm is compared against other algorithms.

In [30], a real VRP has been carried out for the United Kingdom gas delivery industry, where there is a heterogeneous fleet, demand-dependent servicing time, maximum permitted overtime and special low load requirements. Then a mathematical formulation has been created for the problem and optimal solutions have been found for small sized instances. In addition, a new population-based variable neighborhood search algorithm was presented to solve this applied logistic problem, in which for the first time, adaptive memory was combined with the classical iterated memory method.

In [31], a spreadsheet solver is presented for the VRP. This solver was an open-source file that could work for a large number of VRP versions. Then, it was used for two practical applications for health care and tourism industry. Finally, this algorithm was used for standard instances of VRP with capacity and limitation of navigation, and the results showed that this algorithm is very efficient for instances with up to 200 customers and can obtain very good solutions within at most one hour of execution time.

In [32], a new version of the VRP is considered where, for the first time, price-sensitive demands are considered. In this case, the objective is to maximize customer profits. Then, to solve this problem, a mixed mathematical model is presented by commercial software. As it is difficult to use the commercial software for the large sized instances that occurs in the industry and services, the researchers also suggested an effective cutting and page method for the problem. Also, the proposed method may be applied to solve more traditional problems such as routing-locating using non-referable needs and demands or routing a vehicle with price-sensitive demands, in which no previous research has been conducted so far. Their numerical studies demonstrate the fundamental advantage of the integrated model, which can be used to design online systems distribution shacks.

3. Mathematical modeling

From the graph theory point of view, the HFFOVRPTW can be defined as assuming that G(V,A) represents a graph in which V = {0, 1, ..., n} is the set of nodes and A is the set of arcs (edges). If the graph G was not complete, each one would be replaced by an edge with an infinite positive cost. The node 0 represents the warehouse of the goods and the rest of the *n* nodes represent the customers, and each node has the same amount of demand as the q_i product. Also, each arc in A is corresponding to the Euclidean distance. Also, a fleet of K different vehicle types is located at the node 0, so that each vehicle of the k type has Q_k capacity, fixed cost f_k (if the vehicle is used in response to the problem) and the cost varies a_k. In addition to the k*th* type, the number of n_k of the device is available in the fleet. It is to be mentioned that the variable cost is the cost of a distance by a vehicle such that the cost of assessing each arc (i,j) by the k*th* type vehicle is equal $\boldsymbol{c_{ij}^k = c_{ij} \times a_k}$. Therefore, in the HFFOVRPTW, K matrix is symmetric cost and the amount of goods that k*th* vehicle carries when travelling from node i to j node is shown. In addition to the cost for the type k, time is also related to each other. Each customer must be met in its own pre-defined time window, which is limited to the earliest start time e_i and the last service time l_i. In addition, k*th* type vehicles cannot reach a customer after the late service time and if they reach a customer earlier than the start time, they will suffer an extra waiting time w_i^k. Also, for each vehicle type k, t_i^k is the time of entering node i and f_i^k is service time in node i. If the vehicles finish their routes at the maximum allowable travel time r, then the goal in this regard is to find routes for the vehicles with a minimum cost in which all restrictions including capacity, travel time and window restrictions are established for the vehicles. If k*th* vehicle moves directly from node i to node j, $(i,j = 0,1,2,...,n; i \neq j)$, then $x_{ij}^k = 1$ and otherwise $x_{ij}^k = 0$. Therefore, the MILP model of the HFFOVRPTW is as follows.

$$Min \sum_{k=1}^{K} f_k \sum_{j=1}^{n} x_{0j}^k + \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij}^k x_{ij}^k \tag{1}$$

subject to

$$\sum_{k=1}^{K} \sum_{i=0}^{n} x_{ij}^k = 1; j = 1,2,\cdots,n \tag{2}$$

$$\sum_{k=1}^{K} \sum_{j=1}^{n} x_{ij}^k \leq 1; i = 1,2,\cdots,n \tag{3}$$

$$0 \leq \sum_{i=0}^{n} x_{ij}^k - \sum_{i=0}^{n} x_{ji}^k \leq 1; \forall j = 1,2,...,n, \forall k = 1,2,...,K \tag{4}$$

$$\sum_{j=1}^{n} x_{0j}^k \leq n_k; k = 1,2,...,K \tag{5}$$

$$\sum_{k=1}^{K} \sum_{i=0}^{n} y_{ij}^k - \sum_{k=1}^{K} \sum_{i=0}^{n} y_{ji}^k = q_j; \forall j = 1,2,\cdots,n \tag{6}$$

$$q_j x_{ij}^k \leq y_{ij}^k \leq (Q_k - q_i) x_{ij}^k; \forall i,j = 0,1,\cdots,n, i \neq j, \forall k = 1,2,\cdots,K \tag{7}$$

$$\sum_{i=1}^{n} x_{i0}^k = 0; \forall k = 1,2,\cdots,K) \tag{8}$$

$$\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} x_{ij}^k \left(t_{ij}^k + f_i^k + w_i^k\right) \leq r;\ for k \in \{1,2,...,K\} \tag{9}$$

$$t_0^k = w_0^k = f_0^k = 0 \tag{10}$$

$$\sum_{i=0, i \neq j}^{n} x_{ij}^k \left(t_i^k + t_{ij}^k + f_i^k + w_i^k\right) \leq t_j^k; for j \in \{1,2,...,n\} and k \in \{1,2,...,K\} \tag{11}$$

$$e_i \leq \left(t_i^k + w_i^k\right) \leq l_i; for i \in \{1,2,...,n\}\ and k \in \{1,2,...,K\} \tag{12}$$

$$x_{ij}^k \in \{0,1\}; \forall i,j = 0,1,\cdots,n, \forall k = 1,2,\cdots,K \tag{13}$$

$$y_{ij}^k \geq 0; \forall i,j = 0,1,...,n, \forall k = 1,2,...,K \tag{14}$$

In the objective function (1), the first sum shows the total fixed costs of vehicles, and the second sum shows the total variable costs based on the routes travelled by all vehicles. Constraints (2) and (3) show that only one vehicle is imported to each customer and a maximum of one vehicle is return from each customer respectively. Also, the relationship (4) causes the route of each vehicle is continuous from depot to a final customer in which the amount of difference in limit (4) for middle customers in the route is zero, but if these customers are the end nodes, these values are one unit more than the number of exit vehicles. The relationships (5) indicates that the number of vehicles used in the k*th* type should be at most equal to n_k and relationships (6) ensures that each customer should be addressed only in one visit and by a vehicle. Relationship (7) shows that the capacity limitation of each vehicle is observed, and relationships (8) cause no paths exist to the warehouse. The relationship (9) expresses the limitation of maximum travel time and the constraints (10–12) apply the time window limits for each customer. Finally, relationships (13) and (14) determine the range of variables.

4. The proposed algorithm

In general, the algorithms for the VRP are divided into two exact and approximation types. The first category is used for small-scale problems and its optimal solution is obtained at an unacceptable time, but the second category, which has been seriously addressed for almost three decades, can obtains a near-optimal solution of the problem in most cases at an acceptable time. Also, the approximation methods are partitioned into two types - heuristic and *meta*-heuristic methods. In *meta*-heuristics where the CPU time of the algorithm, unlike heuristic methods, depends on the user's decision, the solutions are obtained almost at the close time to heuristic algorithms and less than exact methods. Although the solutions of these methods are usually better than the solutions obtained by heuristic methods and use some strategies for avoiding fall in local optimum point, there are many parameters in these algorithms that need to be obtained experimentally by the user. This large volume of parameters causes algorithms to fail to obtain the same solutions in similar repetitions. Therefore, these algorithms do not have a fixed procedure for achieving the best solution and random parameters play a great role in these algorithms.

One of the most powerful *meta*-heuristic algorithms is the TS algorithm. This method, presented by Glover [33], has evolved in recent years and has become a very efficient method so that in some problems, optimization of compound problems has achieved the best solutions. Recently, this method is enhanced in parallel, using numerous methods to generate neighbors, using numerous methods of diversity and resonance in order to guide the search procedure in the procedure, and using more efficient local search procedures to increase the power of the procedure to find better quality solutions. For obtaining more information about some versions of the algorithm, the reader can refer to [34]. Another modification for this procedure in other literature is to use other methods to combine with this method to further improve the algorithm. For this cause, the primary idea of our proposed algorithm is to deliver a useful algorithm for producing the initial solution needed by the TS algorithm with the sweep (SW) algorithm. Also, different neighbor search algorithms such as insert, exchange and 2-opt algorithms are applied to further search the feasible space and find better solutions. The basic steps of the classic TS algorithm are discussed in Fig. 2.

Max: The defined maximum number of iteration.
X_{best}: An initial solution obtained by another algorithm.
Tlist: The considered tabu list.
N(x): The neighborhood set for solution X.
AC: The aspiration criteria.
Cset: The Candidate set.
I=1;
***While** i<=Max*
    Cset=N(X_{best})-Tlist+AC
    Find the best solution belonging to Cset called $X_{current}$.
    ***If** f($X_{current}$)<f(X_{best})*
        X_{best}= $X_{current}$.
    Update Tlist with FIFO policy.
End

Fig. 2 Pseudo-code of classical TS.

4.1. Generating the initial solutions

The TS algorithm is an improving algorithm that starts with a feasible solution and try to reach better solutions at a reasonable time. Based on the importance of this initial step in our algorithm, the SW algorithm is merged with insert and exchange algorithms. This technique is a very significant part of heuristic methods that can obtain useful solutions of the routing problems quicker than the other methods. This procedure performs in a manner that the customers who have to be served by the vehicles are divided such that they are very close to each other. In other words, this algorithm divides the customers based on geographical areas and each cluster is devoted by a vehicle. The steps of this algorithm for solving the VRP are performed based on the following steps:

5. Step 1: Get the value of polar coordinates for each customer in relation to the warehouse. To get the value of this angle uniquely, consider it relative to the positive axis X and between -pi to + pi.

Step 2: Start from the customer with the smallest angle and meet the customers one by one until the largest amount until the requests of the desired cluster do not exceed the amount of the capacity of the allocated customers' vehicles.

6. Step 3:If it is not possible to add a client to the desired cluster, generate a new cluster starting from step 2.

6.1. Step 4: Continue steps 2 and 3 until all clients are assigned to the clusters.

For this purpose, because the first customer after the warehouse is very important, after finding the order of angles, it is possible to find a new starting point for the algorithm in the same order of the customers with each shift of the customers. Therefore, *n* solutions of the problem can be generated by the algorithm. Now by adding virtual vehicles to the main vehicles, *n* solutions are created. In this step, using two local search algorithms, insert and exchange, it is tried to make these solutions possible in the first step and in the second step, to increase the quality of their objective function as much as possible. In the insert move, a customer is removed from a vehicle path and sent to the vehicle's path provided that the new solution is firstly true in the limitations of the problem and secondly the new solution is better than previous solution (Fig. 3). Also in the exchange move, two customers in the different routes are chosen and exchange their positions (Fig. 4).

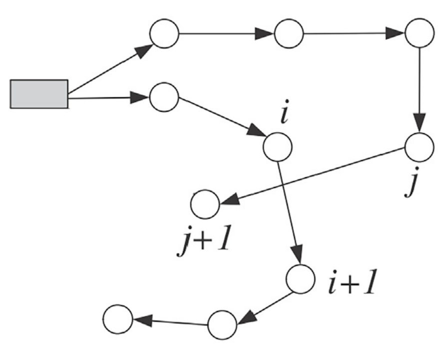

Fig. 3 Example of the insert move.

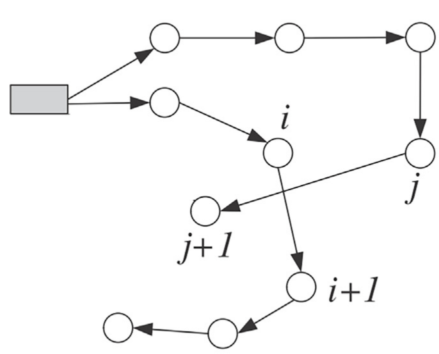

Fig. 4 Example of the exchange move.

Both of these algorithms lead to more search in the search space to improve the best solution obtained by SW algorithm. Any number of n solutions that became possible solutions after using these two algorithms are presented as the initial solutions to the proposed TS algorithm.

6.2. Neighborhood in the proposed algorithm

For every initial solution in the TS algorithm, a neighborhood set is found so that neither its size be large as it is very difficult to calculate feasibility and their objective functions, nor it be small as it does not enhance the solution quality. Now, the solutions of this set are examined and among them the best current solution is selected to compare with the best obtained solution. Many algorithms have been presented so far to create neighborhoods, for example, inserts, exchange, Lin-Kernighan, rearrangement, two-opt, three-opt and k-opt local searches. In the proposed algorithm, two insert and exchange movements for different routes and randomly nearest neighborhood for each Hamiltonian route are used. In other words, the first two movements are applied for changes customers of a route and Lin-Kernighan is used to change its arrangement customers. Because the two insert and exchange operations make many changes in the solution and the feasible space is examined more widely, each one with a 40 % probability per repetition of the algorithm is used. On the other hand, because the Lin-Kernighan algorithm does not change the customers of each vehicle, so the algorithm mostly conducts local search by this algorithm and the neighbors close to the solutions are searched. For this reason, the probability of using this algorithm in the proposed method is considered less and 20 % is selected.

6.3. Tabu list and aspiration conditions

One of the most significant characteristics of the TS algorithm is that unlike the local search, it may be transmitted from a better solution in the present iteration to a worse one in the next iteration based on tabu list. It is to be mentioned that considering worse solution increases the risk of re-observation of a solution and thus creates a cycle. Here, a short-term memory called a tabu list is used to prevent movements that may lead to recurring responses, and all recently observed solutions are listed for a certain number of repetitions. Therefore, the displacement of the current solution to the next one is done when the new solution is not on the list and the aspiration conditions are satisfied in it, which prevents the algorithm from returning to recently observed solutions and creating a round. It is to be mentioned that in the proposed algorithm such as the classic method after each move, the list is updated. In this way, the new solution is added to the list and the oldest one in the list, like the queue, is removed from it. On the other hand, the length of the tabu list controls the search process so that the small size of list causes the tabu mode to be removed faster than the solutions and the search focuses on a small area of the feasible space, but when the size of list is large, more solutions will be listed, and the search range will be expanded. The proposed algorithm uses variable tabu list mode, in which case the length of the list is between the minimum and maximum values. In the process, the length of this list gradually increases from minimum to maximum value. The prime justification for using the variable tabu list in this study is that the process of an efficient *meta*-heuristic algorithm to find a quality solution in a limited time is to search globally at the starting of the algorithm to try to examine the problem space as much as possible. Whenever a good answer is obtained, the algorithm can use local search algorithms to better explore area and come up with better solutions. Therefore, considering the size of the small tabu list at the starting of the algorithm means that the algorithm has full authority to go to the space around each solution, but as it approaches the end of the algorithm, a change in the space search method must be changed from global to local. This method occurs in the proposed algorithm by increasing the length of the tabu list. So, increasing the length of the tabu list causes the algorithm to limit the search for solutions to certain areas and the algorithm only examines limited feasible space.

Another key pillar of the TS algorithm is the aspiration conditions. Although the tabu list causes movement to be banned for a while, this action, in addition to preventing it from a cycle, may also cause the algorithm to lose high-quality solutions. In other words, this makes it easier to manage the list, despite the fact that the search is more limited than the case where all the solutions are saved. Because other solutions that include these features and have not been observed before may also be prohibited. Therefore, to find solution to the problem, these conditions are defined in our proposed algorithm that causes some of the solutions that are on the banned list to be removed from the list sooner than the deadline. The first condition is that if a move to a member leads to a solution with a better cost than the best answer has ever been obtained, then the member is removed from the tabu list. In addition, if the structure of the method and tabu list in the stage of the algorithm is such that there is no possibility of moving to a member. In this case, the member closest to the exit will be removed from the list. It should be noted that these conditions cause the algorithm to escape this problem when it is at a dead end and move towards better solutions.

6.4. *Intensification and diversification mechanisms*

Intensification mechanism is one of the main characteristics of TS algorithm that leads to finding more quality solutions. It is to be mentioned that the search process in the problem space should be able to wisely search for parts of the space that contain higher quality solutions. This perception is executed in the TS algorithm with an intensification process to reinforce the solutions and transitions that could lead to good solutions. In other words, this strategy refers to a return to elite solutions and more searching within their scope. Therefore, it needs the tools to determine the set of elite solutions to have a basis for combining good specialty and creating good new solutions. In our algorithm, this action is executed by altering the obtained routes by the vehicles and is initiated once the best solution is enhanced. It is to be mentioned that there are various means to alter routes traveled by every vehicle, but it is only accepted if it fulfils the restrictions of the problem, stated above, and the new solution will receive a better amount for the problem than the previous one. Besides, another intensification process is using the 2-opt algorithm when the best known solution is enhanced (Fig. 5). The reason why the algorithm is initiated at this phase is that when an improved solution is obtained, it indicates that the algorithm is capable to obtain a better area of possible space, which is possibly very precise to further enhance the solution. Therefore, looking for neighbors of high quality solutions will lead to better solutions if any.

When searching for a solution in the available space, the algorithm may converge prematurely and fail to escape areas with poor quality solutions. For this purpose, in *meta*-heuristic algorithms, a mechanism called diversification is used, which allows the algorithm to move from these areas and search the problem space until the end of the algorithm. In our algorithm for this purpose, the number of iterations is considered and if within this number the algorithm cannot enhance the best solution, this process will be initiated. The procedure is that the algorithm considers the best solution in k of the previous iteration as a criterion and accepts the solution in the current iteration that is better than these solutions. The steps of algorithm are depicted in Fig. 6.

6.5. *Stopping conditions*

Finally, like all metaheuristic methods, the stopping conditions of the algorithm are examined at this stage. The stopping con-

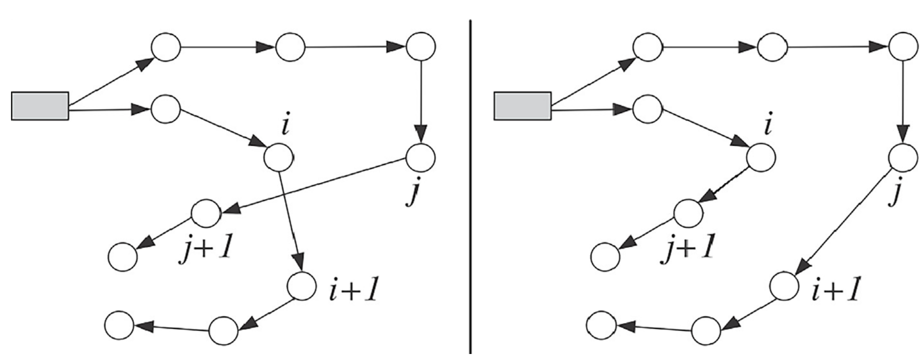

Fig. 5 Example of 2-opt procedure.

```
Input: x as initial solution;
Output: xbest, the best solution found;
Parameters definition
n: number of customers.
Idm: number of iterations in which diversification mechanism is active.
IIm: number of iterations in which diversification mechanism is active.
k: number of iteration which selecting the best solution for diversification mechanism.
Ihbu: maximum number of iterations allowed when BKS has not been updated.
Variable definition
Dvr: It is one the in diversification step, otherwise is zero.
Itf: It is one in intensification step, otherwise is zero.
Initialization
Set Dvr=0, Itf=0, fail-iteration=0, Itf-iteration=0, Dvr-iteration=0, iteration=0.
Set xnow = x, xbest=1E+1000.
Repeat
    Generate Allow-set=N̄(xnow)\TabuList(iteration) + Aspiration(iteration). Choose the
    best solution in Allow-set and set xnow=xbest.
        If ( !Dvr && !Itf )
            If ( c(xnow)≤ c(xbest) )
                xbest = xnow.
                Apply 2-opt algorithm.
                Fail-iteration = 0.
            Else
                If (fail-iteration <Ihbu)
                    Fail-iteration = fail-iteration+1.
                Else
                    Dvr =1.
                    Fail-iteration = 0.
                    Update objective function.
                End if.
            End if.
        End if.
        Else if ( Dvr )
            If ( c(xnow)≤ c(xbest) )
                xbest = xnow.
                Apply 2-opt algorithm.
                If (Dvr-iteration < Idm )
                    Dvr-iteration = Dvr-iteration+1.
                Else
                    Dvr-iteration = 0.
                    Dvr = 0.
                    Itf = 1.
                    Update objective function.
                End if.
            End if.
        End if.
        Else if ( Itf )
            If ( c(xnow)≤ c(xbest) )
                xbest = xnow.
                If (Itf-iteration < Iim )
                    Itf-iteration = Itf-iteration+1.
                Else
                    Itf-iteration = 0.
                    Itf = 0.
                End if.
            End if.
        End if.
        iteration = iteration+1.
Until iteration<=n.
Print (xbest) and its value.
```

Fig. 6 Outline of TS.

```
Consider the obtained feasible solutions found by SW algorithm and
calculate their objective functions.
Repeat
    For each obtained solution in step 1:
        • Create 100 neighbors with the percentage said by three
          algorithms of insertion, exchange and Lin-Kernighan
          algorithms.
        • Select the best solution obtained in the neighborhoods,
          which is not on the tabu list or satisfy in aspiration condition.
        • If no new solution satisfied the conditions, stop, otherwise,
          transfer the new solution to the current solution.
        • Apply intensification mechanism.
        • Put the previous solution on the tabu list and increase the
          length of the list.
Until (the stopping condition is satisfied)
Print the best obtained solution.
```

Fig. 7 Steps of the proposed algorithm.

dition in the proposed algorithm is considered to be the number of customers in each instance. It is to be mentioned that there are other conditions for stopping *meta*-heuristics such as repeating the best solution to a certain number of iterations or implementing the algorithm at a given time those are not considered here.

In Fig. 7, our algorithm is briefly discussed. It is to be mentioned that in this algorithm the same symbols which are defined in section 2 are used. In addition, there are a series of inputs for this algorithm except those mentioned in the proposed algorithm, which is given to the algorithm before the user implements the algorithm, which is as follows.

- Total number of customers with their coordinates.
- The amount of demands and time windows of each customer.
- The total number of vehicles, along with the number of each category.
- Capacity of each category of vehicles.
- Number of repetitions of the best solution ever obtained, m, as the termination criterion of the algorithm.

7. Results and discussion

The proposed algorithm has been encoded in MATLAB and executed on Laptop with corei3, 4 GB RAM and 2.53 GHz specifications. Also, the exact algorithm has been performed using AIMMS software (with Cplex solver). In order to provide a suitable comparison between the proposed algorithm and the exact method, seven small instances including 10–40 customers are considered which are reported in Table 1 along with their characteristics [35]. Also, the specific demands are randomly scattered around the source and the fleet consists of 4 to 18 different vehicles with a specific capacity in which the route length of each truck is not limited. It is to be mentioned that in all tables, the same abbreviations of the second section of the paper have been used. On the other hand, in these instances, fixed cost is not interfered with the calculations and only the variable cost of trucks is considered because in all the articles that have used these instances, the fixed cost of the truck is not considered in the calculations.

In Table 2, the obtained solutions by the proposed ITS algorithm are compared with the solutions by the exact algorithm and classic TS algorithm. In the table, the first and second columns show names and number of customers of every single instance regardless of warehouse, while the third and fourth columns show the solutions and CPU times by the exact algorithm (AIMMS1), respectively. Further, the results of TS algorithm and its CPU time have been reported in the fifth and sixth columns. Additionally, solutions obtained by our ITS algorithm and exact algorithm (AIMMS2) at the same time are reported in the seventh and nineth columns. Finally, the best known solutions (BKS) by any algorithm are reported in the last column. The results indicate that our ITS algorithm has effectiveness compared to AIMMS2 (exact algorithm) and is capable to find better solutions. In other words, the ITS algorithm has obtained much better solutions at the same time compared to the exact method for first three instances. Also, the ITS algorithm has obtained better solutions for the fourth and fifth instances while there is no time restriction. Finally,

Table 1 Characteristics of the first small test set.

Instance	Number of customers	Kind of vehicles	Capacity of vehicles	Fixed cost	Variable cost	Number of each kind of vehicles
1	10	1	20	20	1	1
		2	30	40	1.1	1
		3	40	70	1.3	2
		4	70	200	1.7	1
2	15	1	30	60	1	1
		2	60	100	1.1	1
		3	80	250	1.5	1
		4	150	300	2	1
3	20	1	20	70	1	1
		2	35	120	1.1	2
		3	50	200	1.2	2
		4	120	250	2	3
4	25	1	25	50	1	2
		2	35	80	1.1	2
		3	50	200	1.2	3
		4	120	250	1.7	3
5	30	1	25	35	1	3
		2	35	50	1.1	2
		3	50	75	1.2	4
		4	120	150	1.7	4
		2	120	75	1.1	2
		3	160	100	1.2	3
6	35	1	50	60	0.7	1
		2	120	75	1	2
		3	160	200	1.1	3
		2	140	75	1.7	2
		3	100	225	2.5	2
		4	200	400	2	2
7	40	1	25	20	1	1
		2	35	35	1.1	1
		3	40	50	1.2	3
		4	70	120	1.7	4
		5	100	225	2.5	2
		6	200	400	3.2	1

Table 2 Comparison of results by ITS with exact and TS algorithms.

Instance	N	AIMMS1		TS		ITS		AIMMS2		BKS
		Solution	Time	Solution	Time	Solution	Time	Solution	Time	
1	10	205.32	0.48	205.32	0.84	205.32	0.56	239.14	0.56	205.32
2	15	299.00	135.67	311.25	0.99	299.00	0.95	476.82	0.95	299.00
3	20	412.65	13.15	418.27	1.46	412.65	1.73	512.95	1.73	412.65
4	25	512.11	16.98	503.72	3.86	499.12	3.99	NA	3.99	499.12
5	30	587.76	46.22	562.82	9.16	535.02	6.90	NA	6.90	535.02
6	35	NA	42351.23	395.12	25.12	379.12	15.25	NA	15.25	379.12
7	40	NA	–	688.02	41.92	615.82	30.12	NA	30.12	615.82

for the last two instances, the exact algorithm could not be able to reach a solution, but our ITS algorithm, within just 31 s, could achieve an answerable and possibly optimal solution. In conclusion, it is shown that our ITS algorithm has very good effectiveness to obtain better solutions than that by the traditional TS algorithm and the exact algorithm.

It is to be mentioned that for these instances, as the number of customers increases, the time difference between our ITS algorithm and the exact algorithm increases speedily. Therefore, in daily uses of the problem in services and industries, a suitable quality solution can be obtained by our ITS algorithm quickly. Also, it can be assumed that the ITS algorithm could obtain better solutions within same CPU time than the basic

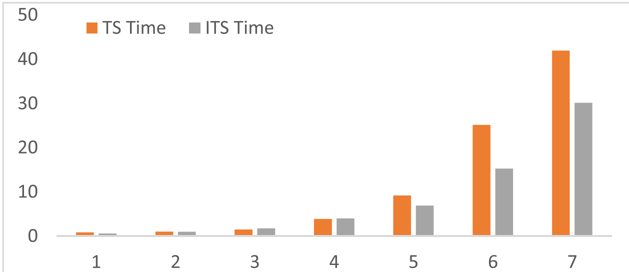

Fig. 8 Comparing CPU time by TS and ITS algorithms for the small sized instances.

TS algorithm. In other words, among these seven instances, the basic TS algorithm and our ITS algorithm have the same solution for the first instance, but the TS algorithm does not have better solutions than the ITS algorithm for the remaining six instances. In addition, comparing the computational times taken by two TS algorithms, one can see that only for third and fourth instances, the time taken by the basic TS algorithm is less than the ITS algorithm, but for the remaining instances, ITS algorithm has been able to obtain the better solutions than the TS algorithm within lesser time (Fig. 8). In the figure, the horizontal and vertical axes show number of instances being tested and the algorithm execution times (in seconds), respectively.

Also, in the columns 7, 8, 9 and 10 of this table, the results of ITS and the exact algorithms are compared from another perspective. Here, the run time is considered same for both algorithms. It is to be mentioned that despite the fact that AIMMS software is very good optimization software, but for sixth and seventh instances, it could not achieve even an answer within the predefined computational time. This comparison shows that if time is considered as the main criterion for comparison between these two algorithms, it is clear that our algorithm has a very good performance to solve this routing problem. More precisely, based on runtime to get the solution, the exact algorithm absolutely is not an effective algorithm to solve the real HFFOVRPTW instances because in real use, the CPU time is the most important goal for the routing and a user cannot delay much time to get an answer.

For further testing of our algorithm, two other categories of instances are considered in this section and the effectiveness of our algorithm is compared with simulated annealing (SA), classic TS and some versions of ITS algorithm. These two categories of instances are randomly scattered around the source, their Euclidean coordinates have been given and their demands are fixed and determined. Also, the fleet consists of several types of trucks with a specified capacity that the length of each

Table 3 Characteristics of the medium instances.

Instance	Number of customers	Kind of vehicles	Capacity of vehicles	Fixed cost	Variable cost	Number of each kind of vehicles
13	50	1	20	20	1	4
		2	30	35	1.1	2
		3	40	50	1.2	4
		4	70	120	1.7	4
		5	120	225	2.5	2
		6	200	400	3.2	2
14	50	1	120	100	1	4
		2	160	1500	1.1	2
		3	300	3500	1.4	1
15	50	1	50	100	1	4
		2	100	250	1.6	3
		3	160	450	2	2
16	50	1	40	100	1	2
		2	80	200	1.6	4
		3	140	400	2.1	3
17	75	1	50	25	1	4
		2	120	80	1.2	4
		3	200	150	1.5	2
		4	350	320	1.8	1
18	75	1	20	10	1	4
		2	50	35	1.3	4
		3	100	100	1.9	2
		4	150	180	2.4	2
		5	250	400	2.9	1
		6	400	800	3.2	1
19	100	1	100	500	1	4
		2	200	1200	1.4	3
		3	300	2100	1.7	3
20	100	1	60	100	1	6
		2	140	300	1.7	4
		3	200	500	2	3

truck's traversable path is not limited. It is to be mentioned that our algorithm for each problem has been run 10 times and the best solution has been shown. In addition, because there is a limitation of service time for customers, therefore, any vehicle with capacity within a certain time period can meet any customer. Also, for a more complete comparison in some cases, the percentage of deviation (PD) to the best solution has been obtained so far, which is derived from formula (15), is reported. In this formula, f(s) represents the solution obtained by one of the algorithms, while BKS represents the best solution ever obtained for that problem. If the percentage of deviation obtained for an arbitrary algorithm is positive, the tested algorithm is worse than BKS and if this value is negative, then the algorithm has been able to increase the quality in terms of obtaining the better solution.

$$PD = \frac{f(s) - BKS}{BKS} \times 100 \tag{15}$$

The characteristics of the second category, which includes 13th to 20th instances, are shown in Table 3 and were first presented by Taillard [35], based on a number of Golden's largest instances, which relate to the FSMVRP with fixed and variable costs, so that in the largest instances instead of fixed cost, the cost is variable. It should be noted that these instances have an average range of customer numbers and vehicles, and the instances with 50–100 customers and 7–18 vehicles are examined. In addition, the values have been chosen so that no truck is much better or worse than other trucks. In other words, if any instance of Taillard is considered as an FSMVRP, the composition of the fleet produced by the algorithm includes all types of trucks. Also, in this algorithm, the number of k*th* vehicles is selected in such a way that in addition to the total capacity of the fleet, the maximum capacity of the smallest truck is more than the total demand of the customers. So, the resulting fleet is different from the fleet of the best solutions to golden instances.

In Table 4, the solutions and CPU time obtained by the proposed ITS algorithm, which includes the best solution (BITS), the worst solution (WITS) and the average solution (MITS) in 10 runs, are compared with the three algorithms - SA, basic TS, and ITS without intensification and 2-opt (ITSWI). In this table, in the first and second columns, the names of the tested instances and the number of customers, are listed respectively, while the third to eighth columns indicate the results by different algorithms. It is to be mentioned that SA, basic TS and ITSWI algorithms have been implemented ten times and their best solutions are reported. In addition, in the ninth column, the BKS has been presented for the instance, which has the same results as the best solution by the proposed ITS algorithm.

By comparing worst, average and best solutions by the proposed ITS algorithm for these medium-sized instances, one can say that the algorithm is very efficient and is stable to solve the instances, and for four instances, 13th to 16th, the results by three versions are equal. As a result, for these instances with a size of 50 customers, there is no need to re-run the algorithm, and in applications one can find a very high-quality solution once the algorithm is executed. On the other hand, for the instances with 75 and 100 customers, there is a little difference between these three versions, but ITS has performed very well and among all algorithms it obtained the best solutions. Also, by comparing ITS with the same algorithm without intensification mechanism (ITSWI), it is found that this mechanism has a great impact on the obtained solutions and all solutions have been improved. On the other hand, the weakest algorithm is the SA algorithm, which was predictable due to its poor strategies for escaping from local optimal points. By comparing the results by SA algorithm with the basic TS algorithm, it can be seen that for the 13th and 16th instances, two algorithms have same solutions, and for the 18th instance, the SA algorithm found better solution than by the TS algorithm. For the remaining five instances, the TS algorithm has found very

Table 4 Comparison of results by proposed algorithms with other metaheuristic algorithms.

Instance	N	WITS		MITS		BITS		BKS
		Solution	Time	Solution	Time	Solution	Time	
13	50	985.72	98.12	985.72	98.23	985.72	99.71	985.72
14	50	448.25	97.10	448.25	97.34	448.25	99.72	448.25
15	50	705.12	95.72	705.12	95.82	705.12	94.82	705.12
16	50	785.82	98.82	785.82	99.28	785.82	99.82	785.82
17	75	832.26	134.82	821.72	136.82	815.05	135.23	815.05
18	75	1652.92	118.82	1611.82	125.82	1595.82	122.82	1595.82
19	100	937.91	195.92	912.81	195.27	890.12	198.82	890.12
20	100	1102.71	201.23	1053.72	204.23	1035.12	199.16	1035.12
Instance	**N**	**TS**		**ITSWI**		**SA**		**BKS**
		Solution	**Time**	**Solution**	**Time**	**Solution**	**Time**	
13	50	1023.61	91.28	1002.71	95.23	1023.61	83.72	985.72
14	50	515.28	92.71	502.81	98.92	536.82	84.82	448.25
15	50	726.84	93.27	712.74	92.11	753.92	81.12	705.12
16	50	814.27	89.02	814.27	93.72	814.27	79.92	785.82
17	75	859.92	123.72	859.92	129.82	873.82	101.27	815.05
18	75	1762.02	119.72	1683.82	117.26	1703.92	103.82	1595.82
19	100	1062.82	179.12	954.82	182.81	1123.82	157.92	890.12
20	100	1106.82	185.63	1073.92	185.02	1173.82	168.62	1035.12

good solutions almost at the same time, that shows its greater capabilities to escape from local optimal points. The point to emphasize here is that the difference in the computational time by all these algorithms, due to the availability of much stronger computers, is negligible and may be possible to obtain these solutions in lesser times.

Further, to observe the performance of ITS algorithm compared to basic TS algorithm, Fig. 9 shows the Gaps compared to BKS for the solutions obtained by each algorithm. In this way, the performance of each algorithm can be seen separately for each instance. In this figure, the horizontal axis shows the names of the tested instances, while the vertical axis shows the Gap value of each algorithm for each instance. As can be seen in this figure, the BITS is mentioned, but because it has the best solution for each instance, it has a zero PD value for all instances. Furthermore, as mentioned above, the classic TS algorithm does not show a good performance to solve these problem instances unless it is modified to further improve. It can also be concluded that the use of a local search mechanism increases the efficiency of metaheuristic algorithms, and it is recommended for each problem to select correct local search algorithm and then to incorporate with the relevant metaheuristic algorithm.

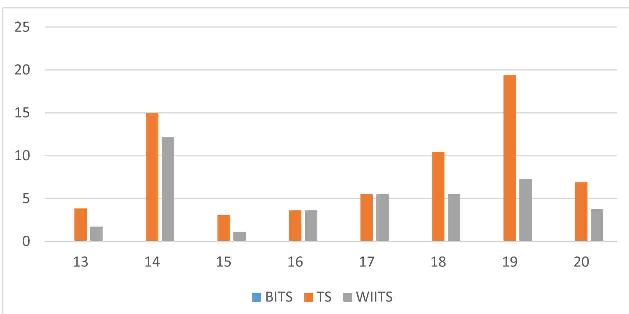

Fig. 9 Comparing Gaps of solutions by TS and ITS (best and worst).

The third category, presented by Li et. al in 2007 [10], has instances of sizes bigger than the 1st and 2nd categories, is shown in Table 5. These instances have number of customers from 200 to 360 and the number of vehicles from 22 to 28. By adding time windows and fixed fleet heterogeneous to this table, the instances of HFFOVRPTW are created.

In Table 6, the comparison among ITS, SA and TS algorithms, is shown on the third category of instances. It is to be mentioned that like the second category, the size of these instances is so high that the exact algorithm could not be applied. According to the results found on these instances with bigger sizes, it seen that ITS algorithm could obtain very good solutions. Also, compared to other algorithms, the ITS algorithm with a best average of 15906.80 produced better solutions than the classic TS algorithm with a best average of 16308.71, the SA algorithm with a best average of 16398.06 and the ITS algorithm without intensification (ITSWI) with a best average of 16087.79. It is to be mentioned that, as with the second category of instances, the TS algorithm has the 2nd

Table 5 Characteristics of the large instances.

Instance	Number of customers	Kind of vehicles	Capacity of vehicles	Fixed cost	Variable cost	Number of each kind of vehicles
H1	200	1	50	20	1	8
		2	100	35	1.1	6
		3	200	50	1.2	4
		4	500	120	1.7	3
		5	1000	225	2.5	1
H2	240	1	50	100	1	10
		2	100	1500	1.1	5
		3	200	3500	1.2	5
		4	500	120	1.7	4
		5	1000	225	2.5	1
H3	280	1	50	100	1	10
		2	100	250	1.1	5
		3	200	50	1.2	5
		4	500	120	1.7	4
		5	1000	225	2.5	2
H4	320	1	50	100	1	10
		2	100	200	1.1	8
		3	200	400	1.2	5
		4	500	120	1.7	2
		5	1000	225	2.5	2
		6	1500	500	3	1
H5	360	1	50	25	1	10
		2	100	80	1.1	8
		3	200	150	1.2	5
		4	500	320	1.7	1
		5	1500	225	2.5	2
		6	2000	250	3	1

Table 6 Comparison of results by proposed ITS algorithm with other metaheuristic algorithms.

Instance	n	WITS		MITS		BITS		BKS
		Solution	Time	Solution	Time	Solution	Time	
H1	200	12762.72	479	12599.92	679	12564.28	476	12564.28
H2	240	11263.82	563	10923.82	557	10783.62	542	10783.62
H3	280	17286.92	713	167526,83	715	16512.82	712	16512.82
H4	320	18124.92	834	17728.90	823	17562.92	812	17562.92
H5	360	22892.92	967	22342.72	957	22110.32	953	22110.32
Mean	–	**16466.26**	**711.2**	**15898.84**	**746.2**	**15906.80**	**699**	**15906.80**
Instance	**n**	**TS**		**ITSWI**		**SA**		**BKS**
		Solution	**Time**	**Solution**	**Time**	**Solution**	**Time**	
H1	200	12652.62	455	12673.82	436	12782.82	401	12564.28
H2	240	11152.72	512	10873.82	512	11372.02	487	10783.62
H3	280	16982.73	698	16872.72	701	17162.83	556	16512.82
H4	320	17872.38	775	17689.55	754	17998.82	678	17562.92
H5	360	22583.11	892	22329.02	901	22673.82	789	22110.32
Mean	–	**16308.71**	**666.6**	**16087.79**	**660.8**	**16398.06**	**582.2**	**15906.80**

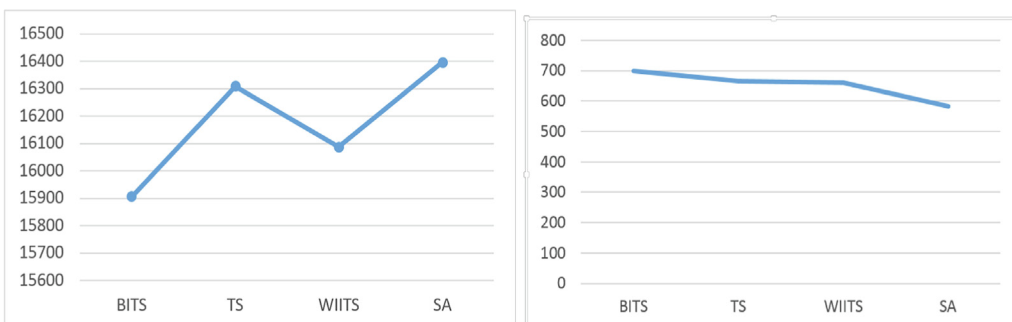

Fig. 10 Comparing the mean solutions by the algorithms.

best results after the ITS algorithm. Also, for clear comparison of all algorithms for these instances, Fig. 10 shows mean obtained results (left figure) and CPU time (right figure). By comparing the execution times of the four algorithms, it is found that although the average execution time of the SA algorithm is better than all algorithms, but solution quality by the algorithm is not good, so, it cannot be introduced as the best algorithm. Further, since the runtime is almost the same for all algorithms, one can say that for large instances it is better to apply the proposed ITS algorithm for real instances.

7.1. Advantage of the proposed algorithm

First, the model based on the MILP has the main advantage of considering the capacity, travel time, the route length and time window constraints, which all are not considered by other models. Next, our proposed ITS algorithm has advantage of considering sweep algorithm along with insert, exchange and 2-opt algorithms to search the feasible space and then to find better solutions. Furthermore, the advantage of using the TS algorithm is that it has the ability to find optimal solutions without getting stuck in repeating the same solution in the next iterations. Finally, one main advantage of our proposed ITS algorithm is that it is flexible, and it can be executed from any initial solution.

8. Conclusion

In this study, a mixed integer linear programming model for the HFFOVRPTW is suggested as one of the newest problems in transportation problems. Then, several standard problem instances were solved by the proposed improved tabu search algorithm. First, the results of the proposed ITS algorithm were compared with the classic TS algorithm and an exact method for small-sized problem instances. The results indicate that our ITS algorithm has effectiveness compared to the exact algorithm and is capable to find better solutions at the same time. It is confirmed that our ITS algorithm has very good effectiveness to obtain better solutions than that by the traditional TS algorithm and the exact algorithm. Next, the results of the proposed ITS algorithm were compared with the classic TS algorithm, SA algorithm and ITS without intensification and 2-opt method for medium-sized and large-sized problem instances. The results indicate that the proposed ITS algorithm is very efficient and is stable to solve the instances. Further, it is found that the intensification mechanism has a great impact on the obtained solutions and all solutions have been improved using this mechanism. Amongst the algorithms, SA is found to be the worst one for both medium-sized and large-sized problem instances. Therefore, due to the difficulty of the problem, for smaller sized problem instances exact algorithms, and for bigger sized problem instances *meta*-heuristic algorithms like our proposed ITS algorithm should be used.

Though our proposed algorithm could obtain very good results, however, for some instances it took long time. So, one can use some effective local search methods and large neighborhoods to solve the problem quickly. Also, other real limitations on this problem such as simultaneous pickup and delivery of goods or the use of balancing can be considered for this problem, and hence, these new problems can be modeled and solved, which will be studied in near future. Further, one can also apply other efficient heuristic algorithms like genetic algorithms, whale optimization algorithm [36] to solve the problem quickly.

Data Availability.

The data used to support the findings of this study are available from the corresponding author upon request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors are very much thankful to the anonymous honorable reviewers for their careful reading of the paper and their many insightful comments and constructive suggestions which helped the authors to improve the paper. The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University for funding this work through Research Group no. RG-21-09-17.

References

[1] M. Ashouri, M. Yousefikhoshbakht, A combination of meta-heuristic and heuristic algorithms for the VRP, OVRP and VRP with simultaneous pickup and delivery, Broad Research in Artificial Intelligence and Neuroscience 8 (2) (2017) 81–95.

[2] M. Yousefikhoshbakht, F. Didehvar, F. Rahmati, An effective rank based ant system algorithm for solving the balanced vehicle routing problem, International Journal of Industrial Engineering 23 (1) (2016) 13–25.

[3] J. Brandão, A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem, European Journal of Operational Research 284 (2) (2020) 559–571.

[4] Shi, Y., Zhou, Y., Boudouh, T., & Grunder, O. (2020). A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window. Engineering Applications of Artificial Intelligence, 95, 103901.

[5] G. Ninikas, I. Minis, The effect of limited resources in the dynamic vehicle routing problem with mixed backhauls, Information 11 (9) (2020) 414.

[6] Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., & Soumis, F., (2002). The VRP with time windows, in: P. Toth D. Vigo (eds.), The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, SIAM, 157-194.

[7] M. Eskandarpour, D. Ouelhadj, S. Hatami, A.A. Juan, B. Khosravi, Enhanced multi-directional local search for the bi-objective heterogeneous vehicle routing problem with multiple driving ranges, European Journal of Operational Research 277 (2) (2019) 479–491.

[8] Z.H. Ahmed, A.S. Hameed, M.L. Mutar, S. Wang, Hybrid Genetic Algorithms for the Asymmetric Distance-Constrained Vehicle Routing Problem, Mathematical Problems in Engineering 2022 (2022) 1–20.

[9] R. Tavakkoli-Moghaddam, M. Meskini, H. Nasseri, H. Tavakkoli-Moghaddam, in: A Multi-depot Close and Open Vehicle Routing Problem With Heterogeneous Vehicles, IEEE, 2019, pp. 1–6.

[10] F. Li, B. Golden, E. Wasil, The open vehicle routing problem: Algorithms, large-scale test problems, and computational results, Computers and Operations Research 34 (10) (2007) 2918–2930.

[11] D. Sariklis, S. Powell, A heuristic method for the open vehicle routing problem, Journal of the Operational Research Society 51 (5) (2000) 564–573.

[12] E. Ruiz, V. Soto-Mendoza, A.E.R. Barbosa, R. Reyes, Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm, Computers & Industrial Engineering 133 (2019) 207–219.

[13] J. Brandão, Iterated local search algorithm with ejection chains for the open vehicle routing problem with time windows, Computers & Industrial Engineering 120 (2018) 146–159.

[14] Y. Xia, Z. Fu, Improved tabu search algorithm for the open vehicle routing problem with soft time windows and satisfaction rate, Cluster Computing 22 (4) (2019) 8725–8733.

[15] A.Z. Şevkli, B. Güler, A multi-phase oscillated variable neighbourhood search algorithm for a real-world open vehicle routing problem, Applied Soft Computing 58 (2017) 128–144.

[16] X. Li, S.C.H. Leung, P. Tian, A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem, Expert Systems with Applications 39 (1) (2012) 365–374.

[17] J. Brandão, A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem, European Journal of Operational Research 195 (3) (2009) 716–728.

[18] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Applied Mathematics and Computation 274 (2016) 292–305.

[19] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, Information Sciences 478 (2019) 499–523.

[20] Z.H. Ahmed, A. Alexandridis, A hybrid algorithm combining lexisearch and genetic algorithms for the quadratic assignment problem, Cogent engineering 5 (1) (2018) 1423743.

[21] Ganji, M., Kazemipoor, H., Molana, S.M.H., & Sajadi, S.M. (2020). A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. Journal of Cleaner Production, 259, 120824.

[22] Y. Yu, S. Wang, J. Wang, M. Huang, A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows, Transportation Research Part B: Methodological 122 (2019) 511–527.

[23] Y. Meliani, Y. Hani, S.L. Elhaq, A. El Mhamedi, A developed Tabu Search algorithm for heterogeneous fleet vehicle routing problem, IFAC-PapersOnLine 52 (13) (2019) 1051–1056.

[24] C.W. Chu, A heuristic algorithm for the truckload and less-than-truckload problem, European Journal of Operational Research 165 (3) (2005) 657–667.

[25] M.C. Bolduc, J. Renaud, F. Boctor, A heuristic for the routing and carrier selection problem, European Journal of Operational Research 183 (2) (2007) 926–932.

[26] C. Prins, Two memetic algorithms for heterogeneous fleet vehicle routing problems, Engineering Applications of Artificial Intelligence 22 (6) (2009) 916–928.

[27] V. Schmid, K.F. Doerner, R.F. Hartl, J.J. Salazar-González, Hybridization of very large neighborhood search for ready-mixed concrete delivery problems, Computers & operations research 37 (3) (2010) 559–574.
[28] J. Euchi, H. Chabchoub, A hybrid tabu search to solve the heterogeneous fixed fleet vehicle routing problem, Logistics Research 2 (1) (2010) 3–11.
[29] J. Brandão, A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem, Computers & Operations Research 38 (1) (2011) 140–151.
[30] L. Simeonova, N. Wassan, S. Salhi, G. Nagy, The heterogeneous fleet vehicle routing problem with light loads and overtime: Formulation and population variable neighbourhood search with adaptive memory, Expert Systems with Applications 114 (2018) 183–195.
[31] G. Erdoğan, An open source spreadsheet solver for vehicle routing problems, Computers & operations research 84 (2017) 62–72.
[32] A. Ahmadi-Javid, E. Amiri, M. Meskar, A profit-maximization location-routing-pricing problem: A branch-and-price algorithm, European Journal of Operational Research 271 (3) (2018) 866–881.
[33] F. Glover, Tabu search—part I, ORSA Journal on computing 1 (3) (1989) 190–206.
[34] J.F. Cordeau, G. Laporte, A. Mercier, A unified tabu search heuristic for vehicle routing problems with time windows, Journal of the Operational research society 52 (8) (2001) 928–936.
[35] É.D. Taillard, A heuristic column generation method for the heterogeneous fleet VRP, RAIRO-Operations Research-Recherche Opérationnelle 33 (1) (1999) 1–14.
[36] Zan, J. (2022). Research on robot path perception and optimization technology based on whale optimization algorithm. Journal of Computational and Cognitive Engineering, 00(00), 1-8. DOI: https://doi.org/10.47852/bonviewJCCE597820205514.