IET Collaborative Intelligent Manufacturing

Research Article

New construction heuristic algorithm for solving the vehicle routing problem with time windows

eISSN 2516-8398
Received on 3rd July 2019
Revised 11th August 2019
Accepted on 21st August 2019
E-First on 25th September 2019
doi: 10.1049/iet-cim.2019.0035
www.ietdl.org

Jun Liu[1], Shuo Feng[1] ✉, Qun Niu[1], Lijuan Ji[1]
[1]*School of Mechanical and Electrical Engineering, Lanzhou University of Technology, Lanzhou, People's Republic of China*
✉ *E-mail: 774632914@qq.com*

Abstract: The vehicle routing problem with time windows (VRPTW) is the most important and widely studied combinational optimisation problem. However, most constructive heuristics create a new path when the customer violates the constraint and cannot insert into any existing path, causing the time window constraint to be tight and generating more paths. Aiming at the above problems, a new constructive heuristic algorithm is proposed. The algorithm firstly uses the convex hull of the customer location to determine the initial seed client and reduces the redundant path; meanwhile, the calculation method of the traditional optimal insertion position is improved, and the calculation efficiency is improved to some extent; in addition, for customers who cannot insert any feasible path, the exchange operator is introduced to give the current solution a disturbance instead of directly creating a new seed client and a new path to further reduce the redundant path. The experimental results show that the algorithm can effectively solve the close time window of VRPTW for evenly distributed customers, and even provide a strict time window for evenly distributed customers and a certain amount of hybrid geographic cluster customers.

1 Introduction

The vehicle routing problem (VRP) is the most important and widely studied combinational optimisation problem. It has become the core issue of the supply chain management and logistic industry, and it is encountered in many practical situations such as postal deliveries, school bus routing and many more. Dantzig and Ramser [1] were the first to study the vehicle routing scheduling problem for gasoline trucks in 1959, and proposed a solution based on linear programming. After that, Clarke and Wright [2] generalised this problem as: how to use a fleet of vehicles to serve a set of customers that geographically dispersed around the depot. In the past few decades, the model of the VRP has been continuously expanded and developed. Linus, Bodin and Golden, Assad, Desrochers and many other scholars have classified the VRP according to different standards or from different perspectives [3]. For example: according to whether there is a time constraint, the VRP can be divided into VRP with time window (which is a specified time interval) and without time window and so on. The typical vehicle routing problem with time window (VRPTW) can be generalised as follows: a fleet of vehicles have to serve a set of geographically dispersed customers. Each customer has a service time, a fixed demand and a time window. Each vehicle has a limited capacity. A set of ordered customers are served by a single vehicle, and the vehicle starts and finishes at the same depot. The objective is to find the appropriate routes that minimises the total travel cost as well as the number of vehicles. The VRPTW is a NP-Complete complete problem [4] whose difficulty lies in how to minimise the total travel cost and the number of vehicles on the premise of meeting the time window. In real life, customers often expect their needs to be met within a specified period of time. Therefore, VRPTW has important practical value.

Since VRPTW has only a small number of instances, it can be solved in an accurate way within an acceptable calculation time. Then a plethora of researchers have focused on heuristic approaches (which seek approximate optimal solutions within reasonable time) for solving the VRPTW, especially for large-scale problem instances [5]. In literature [6], Bräysy comprehensively reviewed the traditional heuristic route construction methods that described the basic features of each method, meanwhile, presented and analysed the experimental results for Solomon's test problems. In addition, many meta-heuristic algorithms are also proposed for solving the VRPTW. Many useful meta-heuristics such as genetic algorithm [7–9], tabu search [10, 11], memetic algorithm [12, 13], fireworks algorithm [14] and so on all have been proposed for the VRPTW. The traditional method generates an initial solution by random arrangement of customers or random selection of seed clients, and uses these random arrangements or seed clients to obtain partial solutions. However, the random solution cannot guarantee the quality of the solution. Excessive randomness will easily violate the constraint. Especially when the time window constraint is tight, it will lead to the generation of redundant paths and increase the operation of eliminating these redundant paths. It will also reduce the planning efficiency of the algorithm. For example, Ombuki proposed a genetic algorithm in literature [15], in initial population creation part, which generated the chromosome by random permutations of *N* customers nodes. Due to the strict time and capacity constraints in VRPTW, the random permutation of customers permits only a small number of customers who can be appended to per route, which incurs redundant routes that need to be eliminated. As a kind of important heuristic method, the construction heuristic method is solved to solve the redundant path problem of the heuristic algorithm to some extent. This method usually selects the client (or arc) to generate the local solution or construct the initial feasible path. The guidelines iteratively insert unserved customers into partial solutions until a viable solution is created, but the initial feasible path is not efficient. Solomon once described several construction heuristic methods for VRPTW in [16], such as savings heuristic, insertion heuristic and sweep heuristic. In addition, Solomon proposed six test problem sets which highlight several factors that can affect the behaviour of routing heuristics. Nowadays, the datasets have become the standard sets for evaluating emerging algorithms to solve the VRPTW.

In general, improvement heuristics modify the initial solution to improve the value of objective function until the target cannot be improved any more. Literature [17] claimed that the basic improvement operators of the VRPTW include movements and exchanges of order segments or customers. The work presented in [18] constructed the neighbourhood solution by some different operators, such as out-exchange, out-relocate, in-exchange and in-relocate based on the initial solution. In the literature [19], the author proposed an improved tuba algorithm and formed the neighbourhood structure by 1-swap operators, 2-opt operators and

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License
(http://creativecommons.org/licenses/by-nc-nd/3.0/)

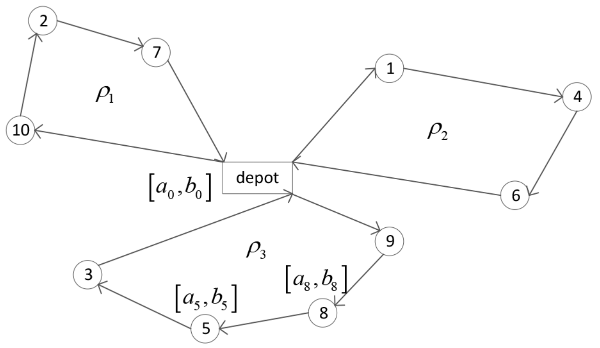


Fig. 1 *Graphical model of the VRPTW*

1-insert operators to extend the exploration space to modify the initial solution. The literature [20] proposed an improved particle swarm optimisation which used a self-adaptive inertia weight and a local search strategy to modify the initial solution and improve the performance of the algorithm. Lzaqebah [21] introduced a modified artificial bee colony (ABC) algorithm. The proposed algorithm improves the results when using a memory by scout bees, where they can memorise the abandoned solutions and select one of these solutions to be replaced by a new generated solution rather than replacing all the abandoned solutions by randomly generated solutions as in the original ABC algorithm.

For the most constructive heuristic algorithm, the initial feasible path is not efficient, and when the customer violates the constraint and cannot be inserted into any existing path, a new path is established, causing more time paths when the time window constraint is tight. A new constructive heuristic algorithm with switching operations is proposed. Motivated by the literature [18], we here introduce a new construction heuristic algorithm which can deal with VRPTW well, especially get an outstanding performance for VRP with tight time windows. The proposed algorithm consists of three phases: initial route construction phase, insertion phase and exchanging operation phase. In the initial route construction phase, different from traditional methods which construct the initial routes by selecting seed customers randomly, the minimal number of vehicles is determined firstly, and then the convex hull of customer locations and the minimal number of vehicles are considered simultaneously to determine the seed customers. The initial routes are constructed by connecting seed customers and the depot, respectively. In the insertion phase, unlike the tradition construction algorithm, two insertion processes are designed to deal with the unassigned customers. In the first process, the unassigned customers are inserted into the initial routes iteratively by best insertion position, where the minimal increment of total travel distance of tours are obtained, and without considering the constraint conditions about vehicle capacity and time windows. In the second process, the unassigned customers who violate the constraints and are removed from the current routes will be inserted into the new feasible routes by considering the free capacity of routes, the demand of customers and the distance between the unassigned customers and feasible routes. In the exchanging operation phase, for unassigned customers who failed in being inserted into any feasible routes, a so-called exchange operator is introduced to the algorithm rather than creating a new route directly as what most of construction heuristics will do. If the exchanging operations fail, the unassigned customer will be selected as a new seed customer to create a new route. The exchanging operator can expand the searching range of the feasible solution by exploring the neighbourhood of the current solution, and avoid generating too many redundant routes. Solomon's benchmark datasets are utilised finally to evaluate the performance of the algorithm and the numerical results show that the proposed algorithm is very efficient in solving instances for uniformly distributed customers with tight time windows, even for uniformly distributed customers mixed with a certain amount of geographically clustered customers with tight time windows. Moreover, the proposed algorithm performs well if the exchanging operation is eliminated while former operations are retained for other instances in R2, RC2, even C proposed by Solomon.

The remainder of this paper is organised as follows. Section 2 provides a description of the VRPTW. Section 3 describes in detail the proposed construction heuristic algorithm. The experimental study performed on standard Solomon's benchmark sets and the corresponding discussion are reported in Section 4. Finally, the conclusions and future work are presented.

2 Problem description

The VRPTW can be formally represented as a directed graph G=(C, E), which consists of a customer set $C = \{0, 1, 2, \ldots, n, n+1\}$ and an edge set E. The node 0 and $n+1$ represent the depot. The set of $n$vertices representing customers is denoted N. Each customer $i(i \in N)$ has a specific demand d_i, a service time s_i and a time window $[a_i, b_i]$, where a_i and b_i are the respective opening time and closing time of customer i. We consider the VRP with hard time window here, so if the vehicle arrives at customer i before the earliest time window a_i, then the vehicle must wait until service is possible, and no vehicle can serve customers past time b_i. The depot also has a time window $[a_0, b_0]$, so the vehicle must return to the depot before or at time b_0. Each vehicle belongs to a homogeneous fleet of K identical vehicles. Each vehicle belongs to a homogeneous fleet of V identical vehicles. Each vehicle has a limited capacity Q. We assume that vehicle velocity equal to 1, according to the formula that the distance is equal to the speed multiplied by the time, the travel distance and travel time at this time are numerically equal. t_{ij} means travel distance of the arc $(i, j) \in E$, the travel time between customers i and $j(i, j \in N)$ is denoted by τ_{ij}. Besides, the C_{ij} represent the cost with each arc $(i, j) \in E$ of the routing network.

There are two decision variables x and s. For each vehicle, the decision variable x_{ijk} is equal to 1 if vehicle k drives from customer i to customer $j(i \neq j, i \neq 0, j \neq n+1)$, and 0 otherwise. The decision variable s_{ik} denotes the time that the vehicle k starts to serve customer i. If vehicle k does not serve customer i, then s_{ik} has no meaning.

Fig. 1 shows a graphical model of the VRPTW and a simple solution. This example has three routes. Depot time window is $[a_0, b_0]$. In route ρ_3, customer 8's earliest and latest service time are respective a_8 and b_8. The starting time of customer 8 adds its service time s_8 and the travel time t_{85}, cannot be later than customer 5's latest service time b_5.

Definition: The ordered sequence of customers visited by a single vehicle k is a route.

According to the description above, the model of VRPTW can be formulated as shown below:

Objective:

$$\min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} C_{ij} x_{ijk} \tag{1}$$

Subject to:

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1, \quad \forall i \in C \tag{2}$$

$$\sum_{j \in N} x_{0jk} = 1, \quad \forall k \in K \tag{3}$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0, \quad \forall h \in C, \forall k \in K \tag{4}$$

$$\sum_{i \in N} x_{i,n+1,k} = 1, \quad \forall k \in K \tag{5}$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ijk} \le Q, \quad \forall k \in K \tag{6}$$

$$s_{ik} + t_{ij} - (1 - x_{ijk})k \le s_{jk}, \quad \forall i \in N, \forall j \in N, \forall k \in K \tag{7}$$

$$a_i \le s_{ik} \le b_i, \quad i \in N, k \in K \tag{8}$$

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License
(http://creativecommons.org/licenses/by-nc-nd/3.0/)

$$s_{ijk} \in \{0, 1\}, \quad i \in N, j \in N, k \in K \tag{9}$$

The objective function (1) is to minimise the number of homogeneous vehicles and the total travel distance. In real life, the objective of minimising the fleet size is more important than minimising the total travel distance because the vehicle-related cost is generally higher than the cost of travel distance. Therefore, a solution requiring fewer vehicles is considered better than a solution with more routes, regardless of the total travelled distance. Let λ_a and λ_b be two solutions to the VRPTW. Then the quality of the solution λ_a is higher, if $K(\lambda_a) < K(\lambda_b)$ or ($K(\lambda_a) = K(\lambda_b)$ and $T(\lambda_a) < T(\lambda_b)$).

The constraint set (2) states that each customer must be visited by a single vehicle at most. The constraints (3), (4), (5) give the flow constraints that ensure each vehicle leaves from depot 0 and finally returns to the same depot after visiting corresponding customers. The constraint set (6) guarantees that the sum of customer demands assigned to a vehicle should not exceed the vehicle capacity Q. The constraint set (7) states that if the vehicle travels from customer i to j, it cannot arrive at customer j before $s_{ik} + t_{ij}$. The constraint set (8) ensures that time windows are observed by all of the routes. The constraint set (9) gives the set of integrality constraints.

3 New construction heuristic algorithm

This section presents the proposed construction algorithm in detail. The algorithm consists of three phases. In the first phase, we set the number of seed customers or initial routes r by the minimal number of vehicles and select r geographically dispersed customers as the seed customers to construct initial routes. In the second phase, the unassigned customers are firstly inserted into the initial routes iteratively by minimising the increment of the total travel distance and without considering the constraints. Then started from the depot, each customer is examined one by one by the constraint conditions of the vehicle capacity and the time windows. For those who violate the constraints, remove them from the current route to get new feasible routes and reinsert them into the new feasible routes again by considering the free capacity of routes, the demand of customers and the distance between the unassigned customers and feasible routes. In the third phase, for reinserted customers who violate the constraints, an exchanging operator is involved in to make them be feasibly assigned. If the exchanging operation fails, the unassigned customer will be selected as a new seed customer to create a new route.

3.1 Initial route construction phase

At the beginning of this process, we need to decide the number of initial routes 'r' to construct routes. VRPTW is a discrete combination optimisation problem, it has two objectives. These objectives affect one another in non-linear ways: small vehicle number may obtain higher travel distance, but more vehicles may have smaller or greater travel distance. As previously stated, the fixed cost of vehicle is greater than its travel cost. We regard the objective to minimise the size K of the fleet as the primary goal. It is easy to find that $K \geq K_{\min}$, we can calculate the number of ideal minimal routes by (10) as follows:

$$K_{\min} = \frac{D}{Q}\left(D = \sum_{i=1}^{n} q_i\right) \tag{10}$$

where D represents the total customer demands. Q is the limited capacity of each vehicle.

It is apparent that the effort to remove the redundant routes in traditional construction algorithms is to some extent reduced if the initial customers is determined based on the number $r = K_{\min}$.

Moreover, considering the maximum distance from the depot, seed customers can be selected by the convex hull of the customer locations. For more help about the selection method, please refer to the literature [22]. Let L_{ch} be the set of the customers on the convex hull and $L_{\overline{ch}}$ be the set of the inner customers of the convex hull. Especially, the customers in $L_{\overline{ch}}$ are arranged in ascending order of $b_i - \tau_{0i}$. Small value of $b_i - \tau_{0i}$ represents the low flexibility of the assigned customer i. This implies that the closing time of the time window for customer i is close to the time required to travel from the depot to customer i. Let S denote the set of assigned customers and $\overline{S}$ denote the set of unassigned customers. The initial route construction process is described as follows:

(i) Construct two lists L_{ch} and $L_{\overline{ch}}$. And set S to be empty.

(ii) Select a customer $i \in L_{ch}$ who is the one with the farthest distance from the depot as the first seed customer. Let $S = S \cup \{i\}$ and built the first initial route by connecting the seed customer i and the depot.

(iii) Find the customer $j \in L_{ch}$ whose position maximises the sum of distances between the customer j and the existing seeds. Calculate the minimum distance d_j between the customer j and the existing seed customers. Meanwhile, calculate the minimum distance d_i between the first customer $i$of $L_{\overline{ch}}$ and the existing seeds.

(iv) The customer who is with max $\{d_j, d_i\}$ is selected as the next seed customer. It will make that the seed customers are geographically dispersed as large as possible. Build a new route with the new seed customer. Update L_{ch}, $L_{\overline{ch}}$, $S$and $\overline{S}$.

(v) Repeat steps (3) and (4) until the number of seed customers in S reaches to $K_{\min}$.

3.2 Insertion phase

The initial routing construction phase is followed by an insertion phase which inserts other unassigned customers in $\bar{S}$ into the initial routes iteratively by minimising the increment of the total travel distance of tours and without considering the constraints until each customer is assigned, i.e. $\overline{S} = \emptyset$.

For traditional construction heuristics, most researchers firstly take into account whether the vehicle capacity and time window constraints are violated when each customer is assigned to the current routes. If the constraints are violated, then the customer will be treated as a seed customer. This may bring too many redundant routes which should be eliminated in following process owing to tight constrains if the VRP is with tight time window. Nevertheless, two insertion processes are designed to deal with the unassigned customers. In the first process, the best insertion location is considered and determined at first in the paper. After that, an examination and adjustment process are performed. In the second process, the unassigned customers who violate the constraints and are removed from the current routes will be inserted into the new feasible routes by considering the free capacity of routes, the demand of customers and the distance between the unassigned customers and feasible routes.

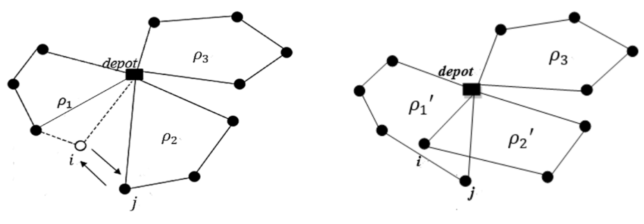
For each of the unassigned customers in the paper, the best insertion position between two consecutive customers in one route is determined according to the following equation:

$$\min I = t_{ip} + t_{pj} - t_{ij} \tag{11}$$

where $p \in \overline{S}$ is the unassigned customer, which will be inserted into two consecutive customers i and $j(i, j \in S)$. Then I denotes the increment of travel distance due to the insertion of the customer p. Note that in most researches such as the literature [15], the best insertion location is determined by the method of minimising the total travel distance of tours, in which the total travel distance of tours will be calculated over and over again. Nevertheless, the increment of the travel distance considered in the paper will bring small calculations.

Then started from the depot, each customer is orderly examined one by one by the constraint conditions of the vehicle capacity and the time windows, i.e. examined by (6) and (8). The customer who violates the constraints will be removed from the current route and also be removed from S to $\overline{S}$, until a new feasible tour is obtained. The feasible routes are ranked by the free capacity of routes in descending order. Meanwhile, the customers in $\bar{S}$ are arranged by

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by-nc-nd/3.0/)



a *b*

Fig. 2 *Operation of the exchange operator*
(a) Before cross operation, ***(b)*** After cross operation

the demand $d_i, (i \in N)$ in descending order. Then in the following process, the first customer i of $\bar{S}$ is selected to be reinserted into the feasible route that has enough free capacity of routes and is nearest to the customer i. Moreover, the insertion location for the customer i is between the two customers who are in the feasible route and are nearest to the customer i. Then another examination is adopted to check whether the customers meet the constraints. If they do, a new feasible route is obtained and the customer i is removed from $\bar{S}$ to S. Otherwise, an exchange operator offered in the next phase is introduced to the algorithm if all current feasible routes are failed to be successfully inserted. For other customers of $\bar{S}$, the same processing is adopted.

3.3 Exchanging operation phase

For the customer i who cannot be successfully inserted in any current feasible routes in insertion phase, an exchanging operator $\in$ is then introduced to the heuristic algorithm. $\in (i, j)$ means the operation of exchanging the locations of the customer i and the customer j.

Firstly, the feasible route which is not the one obtained by removing the customer i from his best insertion location and is nearest to the customer i is selected. Secondly, the customer j who is nearest to the customer i in the selected feasible route is selected firstly and $\in (i, j)$ is taken. Thirdly, the constraints of (6) and (8) about time window and vehicle capacity are examined in both routes. If the constraints are satisfied in both routes, the customer i is removed from $\bar{S}$ to S and the next customer f with the largest demand in $\bar{S}$ is selected to reinsert. Otherwise, the corresponding customers in other feasible routes are selected to exchange locations with the customer i by $\in$. Fourthly, doing that so forth until all feasible routes are tried or a limited number of exchanging operations is reached to. If the exchanging operation does not succeed for the customer i at last, then the customer is handled as a seed customer and a new feasible route is created by connecting the depot and the customer i. The procedure above is iteratively performed until all the unassigned customers have been arranged, i.e. $\bar{S} = \varnothing$.

The operation of the proposed exchange operator $\in (i, j)$ is as shown in Fig. 2. The best insertion location of the customer i has been determined as shown in Fig. 2*a*. However, the customer i did not pass the examination about the constraints. The customer i is removed from S to $\bar{S}$ and one feasible routes ρ_1 is obtained in the insertion phase. ρ_2 and ρ_3 are also two feasible routes. If the customer i has failed to be inserted into ρ_2 and ρ_3, then the customer $j \in S$, who is nearest to the customer i, is selected to take $\in (i, j)$ and two new routes ρ_1' and ρ_2' are obtained as shown in Fig. 2*b*. A following examination as mentioned above is taken again.

Note that a limited number of exchanging operations is considered in fourth step in case of excessive calculation. Many constraints are tangled together for the VRPTW. If the size of the problem is big enough, too many exchanging operations may result in too excessive calculation. Meanwhile, the exchange operator is in fact proposed to give a small perturbation and expand the searching range of the feasible solution. The closer one customer is to another, the bigger the probability of the successful exchanging is. Consequently, for customers who are one another with long distance, the exchanging operation does not work well. Moreover, by our tests in practice, 20–25 times of the exchanging operation for a customer in $\bar{S}$ seem to be feasible for problems with the size of about 100 customers.

Traditionally, most of methods create a new route when the unassigned customer cannot be inserted in any existing routes due to constraints violation. It is easy to create the redundant routes when the time constraints are tight. The paper employs an exchange operator ε to probably convert an unfeasible partial solution (there are still unassigned customers) into a feasible partial one. This operator gives a perturbation and offers a way to expand the searching scope around current solutions, and then we may probably obtain a feasible solution in the current solution's neighbourhood rather than create a new route directly. This way can avoid bringing too many redundant routes or redundant vehicles in practice.

3.4 Algorithm steps description

(1) Initialisation.
(2) Determine the minimum number of vehicles by (10). Then, construct the initial routes by the method described as that in initial route construction phase.
(3) Insert the unassigned customer into the initial routes by utilising the best insertion location mentioned in insertion phase.
(4) Check the customers one by one by the constraint conditions of the vehicle capacity and the time windows. Meanwhile, remove the customers who violate the constraints from S to $\bar{S}$.
(5) Let $f = 1$. Rank the customers in $\bar{S}$ by the demand d_i in descending order. Meanwhile, get the number U of customers in $\bar{S}$.
(6) Let $g = 1$. Determine whether there are the feasible routes which have enough free capacity of vehicles to insert the f customer of $\bar{S}$. If they exist, rank the feasible routes by the distance from the f customer in descending order and get the number V of the feasible routes. If they do not exist, then if $f < U, f = f + 1$ and go to (6), if $f \geq U$, go to (8).
(7) Insert the f customer into the g feasible routes and check the constraints. When the customers meet the constraints, remove the f customer from $\bar{S}$ to S, renew the feasible routes and unassigned customers in $\bar{S}$, $U = U - 1$, then if $U \geq 1$ go to (6), or go to (8). When the constraints are not satisfied, then if $g < V$, $g = g + 1$ and go to (7), if $g \geq V$, $f = f + 1$ and go to (6).
(8) Determine whether $\bar{S} = \varnothing$ exists or not. If it exists, go to (9) or the exchanging operation described as that in exchange operation phase is taken for the next customer in $\bar{S}$, go to (8).
(9) End (Fig. 3).

4 Complexity analysis of the algorithm

Assume that the maximum number of customers is denoted by $n$and $O(\cdot)$ denotes the time complexity function of the proposed algorithm. As the basic operation is to select seed customer in initial route construction phase and the number of initial customers equal to $K_{\min} = D/Q(D = \sum_{i=1}^{n} q_i)$, the maximum construction number of initial routes is $C_n^{K_{\min}}$. That is to say, the time complexity of the proposed algorithm in the initial route construction phase is $O(n^2)$.

In the following, there are three basic operation processes. Firstly, the unassigned customers are inserted into the initial routes by minimising the increment of the total travel distance of tours and without considering the constraints. The best insertion position of all unsigned customers may be determined after calculations of $\sum_{i=0}^{n-K_{\min}+1} C_{n-K_{\min}-i}^{1} C_{2K_{\min}+i}^{1}$. The time complexity of the proposed algorithm in the basic operation process is apparently $O(n^3)$. Secondly, the customers of $(n - K_{\min})$ are one by one examined by the constraint conditions. The time complexity in the operation process is $O(n)$. Thirdly, the customers who violate constraints will be removed from the current route and reinsertion operation processes are following adopted. For those who cannot be

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License
(http://creativecommons.org/licenses/by-nc-nd/3.0/)

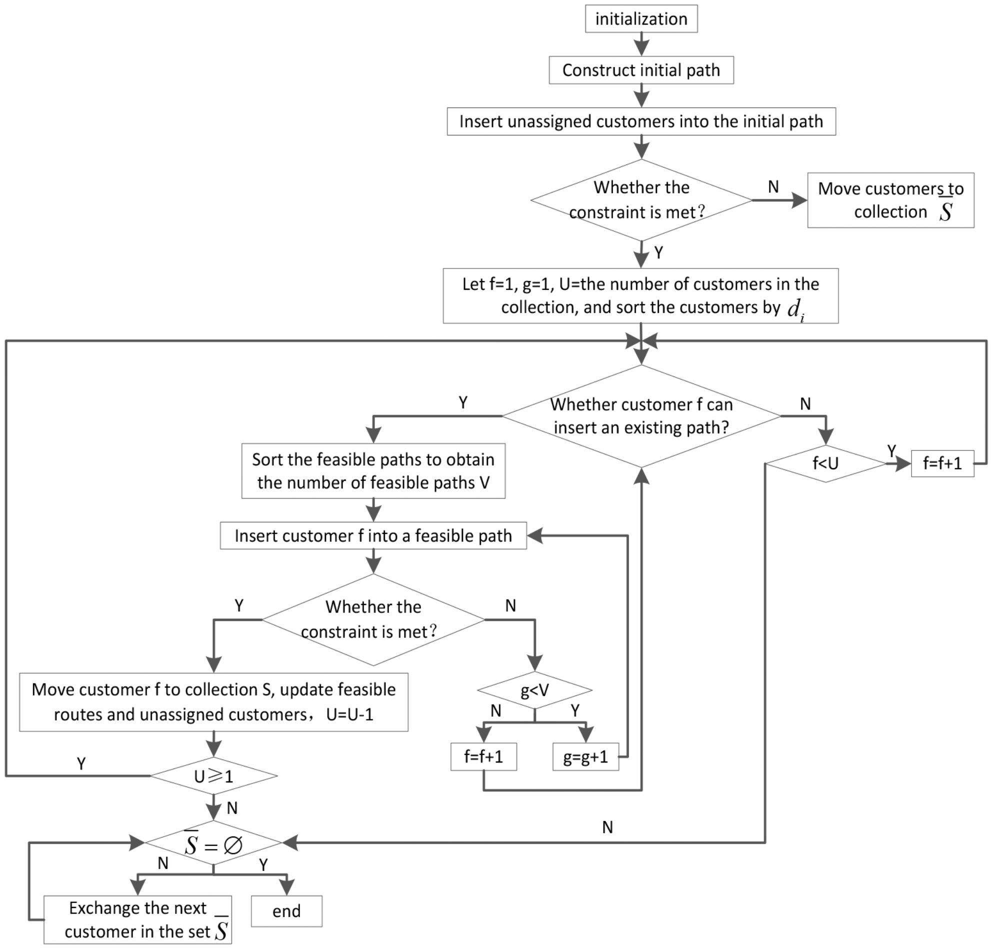


Fig. 3 *Algorithm flowchart*

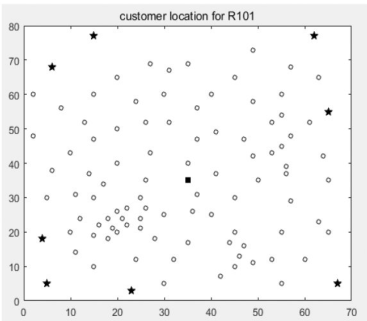


Fig. 4 *Customer location for R101*

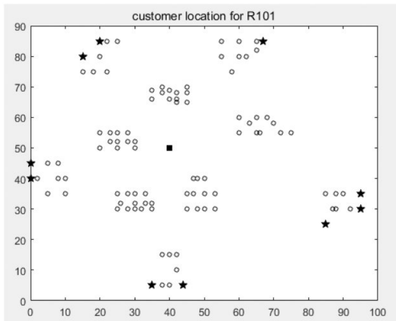


Fig. 5 *Customer location for C101*

reinserted successfully, the exchanging operations $\in (i, j)$ are then executed. The worst one is considered, i.e. the customers of $(n - K_{\min})$ are all to be reinserted and taken exchanging operation. It is apparent that the time complexity $O(\cdot)$ in the operation process is $O(n^3) \cdot O(n) = O(n^4)$. Considering the whole operation process, the time complexity of the proposed algorithm is $O(n^4)$. The efficiency of the proposed algorithm is rather satisfied.

5 Experimental results

5.1 Setup and benchmark instances

In order to evaluate the performance of the proposed algorithm, experiments are implemented by Matlab R2010a on a computer equipped with an Intel(R) Core i3-4130 CPU, 4 GB, Windows 7 operating system.

The proposed construction heuristic algorithm for VRPTW was tested on the classical benchmarks proposed by Solomon [23]. Solomon's benchmarks became the standard sets for evaluating the quality of algorithms used for VRPTW since they reflect various real-life scheduling circumstances. Solomon's datasets consist of six clusters: R1, R2, C1, C2, RC1, RC2. These datasets design highlights several factors that affect the behaviour of routing and scheduling algorithms, they are: geographical data; the number of customers serviced by a vehicle; percent of time-constrained customers; and tightness and positioning of the time windows. Problems in the C category mean the geographical data are clustered, problems in category R mean the customer locations are uniformly distributed whereas those in category RC imply hybrid problems with mixed characteristics from both R and C. In addition, for R1, C1, RC1 problem sets, the vehicle capacity is small and the time window is narrow and allow only a few customers per route (∼5 to 10); in contrast, the sets R2, C2 and RC2 have a large vehicle capacity and a long scheduling horizon permitting many customers (more than 30) to be serviced by the same vehicle. The tested problem has 100 customers where travel times equal to the corresponding distances.

In Figs. 4–6, we illustrate some of the typical customer location examples. Dots in figures denote the customers that need to be served, rectangle represents the depot and stars denote the initial

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by-nc-nd/3.0/)

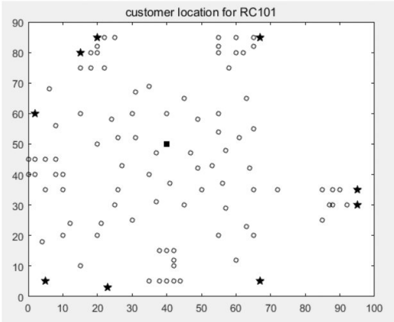


Fig. 6 *Customer location for RC101*

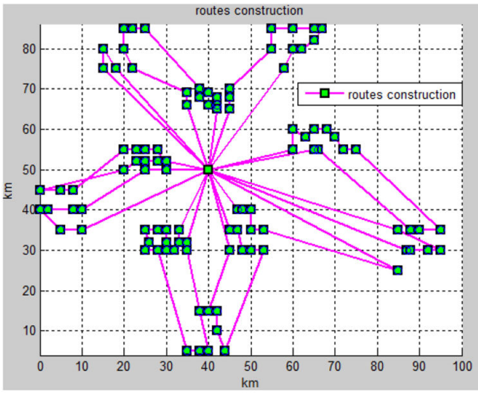


Fig. 7 *Routes of C101 without evaluating constraints*

selected seed customers. As shown in these figures, the seed customers have strong geographical dispersion; they are basically in the convex hull set of customer locations.

5.2 Comparisons with best published results and analysis

We compared the result of the proposed algorithm with the best currently known result published at the SINTEF website [24]. Some comparison results are shown in Tables 1–3. It noted that the numerical examples in the paper are only a few among those we tested.

It also should be noted that solutions in which customers are served by a larger number of vehicles are usually characterised by a shorter total travel distance. However, the first priority of the VRPTW in practice is to minimise the number of vehicles, thus the solution with a small vehicle number is better than the solution with a large fleet size.

For the VRP with uniformly distributed customers, tight time window and smaller vehicle capacity, i.e. for R1, the solutions obtained by the proposed algorithm are better than the published best solutions. This is not only drawn by the results for R101, R102, R103 as well as R105, but also by the results for R104. The vehicle numbers obtained by the proposed algorithm for R101, R102, R103, R105, RC102 as well as RC104, are all smaller than the published best solutions.

This is because our algorithm determines the initial feasible routes by the minimal number of vehicles and the best insertion locations for other unassigned customers are considered firstly, meanwhile, the exchanging operator is utilised to convert a partially unfeasible solution into a feasible one rather than creating a new seed customer or a new route directly when the constraints are violated in insertion phase. All the matters are helpful to avoid generating redundant routes. However, if the uniformly distributed customers are mixed with geographically clustered customers, the superiority of the proposed algorithm will decline with the increasing percentage of geographically clustered customers. This can be seen from the VRPs in RC101, RC103 as well as RC105: the solutions obtained by the proposed algorithm have longer travel distance than what the published best solution has although their vehicle sizes are equal. The coordinates of clustered customers are relatively concentrated (e.g. Fig. 5), so the customers who belong to the same cluster always are in same feasible route. That is to say, the collection of feasible routes is basically as same as the original collection of customer clusters (e.g. Fig. 7). So, the exchanging operation in the paper may disrupt the original customer clusters and create an increment of total travel distance, or even generate redundant routes due to constraints violation. However, if the percentage of geographically clustered customers is small or the time windows is relatively loose like RC102 or RC104, the proposed method still performs rather well. Then our method is not good for the VRPTW in the C category for same reason. Table 3 gives some numerical examples. It is apparent that the fleet sizes obtained by the proposed method are larger than the published best solutions.

Meanwhile, for VRP with wide time window and large vehicle capacity, the solutions obtained by the proposed method are in the mass inferior to the published best solution as shown as in Table 2. This can be expected. The wider time window and larger vehicle capacity means more customers who can be inserted into one

Table 1 Comparison results for R1 and RC1 sets with tight time window and small vehicle capacity

Instance	Best result (vehicle number)	Best result (distance)	Proposed (vehicle number)	Proposed (distance)
R101	19	1650.80	18	1706.5
R102	17	1486.12	15	1523.9
R103	13	1292.68	11	1374.2
R104	9	1007.31	9	977.8
R105	14	1377.11	13	1634.4
RC101	14	1696.95	14	1882.1
RC102	12	1554.75	11	1962.7
RC103	11	1261.67	11	1267.1
RC104	10	1135.48	9	1897.5
RC105	13	1629.44	13	1743.8

Table 2 Comparison results for R2 and RC2 sets with wide time window and large vehicle capacity

Instance	Best result (vehicle number)	Best result (distance)	Proposed (vehicle number)	Proposed (distance)
R201	4	1252.37	6	1604.9
R202	3	1191.7	4	1300.1
R203	3	939.5	3	983.8
R204	2	825.52	2	851.2
R205	3	994.43	4	1138.9
RC201	4	1406.94	6	2004.0
RC202	3	1365.65	6	1550.5
RC203	3	1049.62	4	1076.9
RC204	3	798.46	3	775.83
RC205	4	1297.65	4	1388.6

Table 3 Comparison results for C sets

Instance	Best result (vehicle number)	Best result (distance)	Proposed (vehicle number)	Proposed (distance)
C101	10	828.94	13	1545.7
C102	10	828.94	12	1649.5
C103	10	828.06	13	1532.5
C104	10	824.78	12	1379.2
C105	10	828.94	12	1513.1
C201	3	591.56	5	962.4
C202	3	591.56	7	1117.3
C203	3	591.17	5	912.8
C204	3	590.60	6	1017.4
C205	3	588.88	6	981.08

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License
(http://creativecommons.org/licenses/by-nc-nd/3.0/)

Table 4 Comparison results for R2 and RC 2sets algorithm without exchanging operation

Instance	Best result (vehicle number)	Best result (distance)	Proposed (vehicle number)	Proposed (distance)
R201	4	1252.37	4	1593.1
R202	3	1191.7	3	1290.2
R203	3	939.5	3	983.8
R204	2	825.52	2	851.2
R205	3	994.43	3	1109.2
RC201	4	1406.94	4	1719.7
RC202	3	1365.65	3	1393.7
RC203	3	1049.62	3	1085.9
RC204	3	798.46	3	796.4
RC205	4	1297.65	4	1919.4

feasible route, i.e. smaller fleet sizes are obtained. If the exchanging operator is employed, that probably results in one of the assigned customers to be exchanged with another farther unassigned customer although the new obtained routes may be still feasible. That probably brings longer travel distance even redundant routes. For instance, Table 2 shows some numerical results. It is apparent that the travel distances obtained by the proposed algorithm are longer than what the published best solutions have although their vehicle sizes are equal for R203,R204, RC204 and RC205. Meanwhile, the fleet sizes obtained by the proposed method are larger than what the published best solutions have for R201, R202, RC201, RC202 and so on. This explanation can be testified to some extent by the experimental results in Table 4. Table 4 shows that the proposed method gets results nearly as good as the published best solutions if the exchanging operation is eliminated while former operations are retained. Considering the priority of minimising the vehicle size, the reinsertion position of the customer who violates the constraints is determined by firstly examining the route that is with maximum remaining capacity rather than the shortest distance away from the customer. The new obtained route may be feasible due to the wide time window. However, that inevitably brings about the increment of travel distance. That is why the travel distances obtained by the proposed method are larger than what the published best solutions have although their fleet sizes are equal in Table 4. For the VRPTW in the C category, the similar conclusions can be drawn if the exchanging operation is eliminated while former operations are retained. For concision, the numerical experiments for C category are omitted here.

The numerical experiments we tested confirm that the proposed algorithm is very efficient in solving instances for uniformly distributed customers with tight time windows (R1), even for uniformly distributed customers mixed with a certain amount of geographically clustered customers with tight time windows (RC1). In practice, we are mainly faced with the VRPs with uniformly distributed customers or uniformly distributed customers mixed with geographically clustered customers. Moreover, the key issue or the main difficulty of the VRP is to deal with customers with tight time windows. The significance of the proposed method to some extent lies in it. That is to say, the proposed method has a wide application space in practice. Moreover, it also performs well if the exchanging operation is eliminated while former operations are retained for those in category R2, RC2, even C.

6 Conclusion

In the paper, a new construction heuristic algorithm is proposed for solving the VRPTW. The minimal number of vehicles and the convex hull of customer locations are utilised to determine the initial customers. Meanwhile, two insertion processes and a so-called exchanging operator are introduced to the algorithm. The numerical experiments show that the proposed algorithm is extremely efficient for uniformly distributed customers with tight time windows, even for uniformly distributed customers mixed with a certain amount of geographically clustered customers with tight time windows.

Moreover, the proposed algorithm performs well if the exchanging operation is eliminated while former operations are retained for those in R2, RC2, even C proposed by Solomon. However, we still need to pay more attention at the optimisation of travel distance to reduce greenhouse gas emissions. Hence, we will further try to address this issue on the basis of the work.

7 Acknowledgment

The authors would like to acknowledge the National Natural Science Foundation of China (Grant Number 71861025), National Key Research and Development Plan (Grant Number 2018YFB1703105) and Hongliu First-class Disciplines Development Program of Lanzhou University of Technology for funding this research.

8 References

[1] Dantzing, G., Ramser, J.: 'The truck dispatching problem', *Manage. Sci.*, 1959, **10**, (6), pp. 80–91
[2] Clarke, G., Wright, J.R.: 'Scheduling of vehicle routing problem from a central to a number of delivery points', *Oper. Res.*, 1964, **12**, pp. 568–581
[3] Li, J., Guo, Y.: '*Theory and method of optimized dispatching of logistics distribution vehicle*' (China Materials Publishing House, Beijing, 2001)
[4] Garey, M.R., Johnson, D. S.: '*Computers and intractability: A guide to the theory of NP-completeness*' (W.H. Freeman and Company, USA, 1979)
[5] Lenstra, J.K., Kan, A.H.: 'Complexity of vehicle routing and scheduling problems', *Networks*, 2010, **11**, (2), pp. 221–227
[6] Bräysy, O., Gendreau, M.: 'Vehicle routing problem with time windows, part I: route construction and local search algorithms', *Transp. Sci.*, 2005, **39**, (1), pp. 104–118
[7] Kaabi, H., Jabeur, K.: 'Hybrid algorithm for solving the multi-compartment vehicle routing problem with time windows and profit'. International Conference on Informatics in Control, Automation and Robotics IEEE, Colmar, Alsace, France, 2015, pp. 324–329
[8] Lozano, M., Herrera, F., Cano, J.R.: 'Replacement strategies to preserve useful diversity in steady-state genetic algorithms', *Inf. Sci.*, 2008, **178**, (23), pp. 4421–4433
[9] Mohammed, M.A, Gani, M.K.A., Hamed, R.I., *et al.*: 'Solving vehicle routing problem by using improved genetic algorithm for optimal solution', *J. Comput. Sci.*, 2017, **21**, pp. 255–262
[10] Liu, R., Xie, X., Augusto, V.: 'Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care', *Eur. J. Oper. Res.*, 2013, **230**, (3), pp. 475–486
[11] Liu, R., Xie, X., Garaix, T.: 'Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics', *Omega. (Westport)*, 2014, **47**, (3), pp. 17–32
[12] Labadi, N., Prins, C., Reghioui, M.: 'A memetic algorithm for the vehicle routing problem with time windows', *RAIRO – Oper. Res.*, 2008, **42**, (3), pp. 415–431
[13] Qi, Y., Hou, Z., Li, H.: 'A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows', *Comput. Oper. Res.*, 2015, **62**, (C), pp. 61–77
[14] Abdulmajeed, N.H., Ayob, M.: 'A firework algorithm for solving capacitated vehicle routing problem', *Int. J. Adv. Comput. Technol.*, 2014, **6**, (1), 79–86
[15] Ombuki, B., Ross, B.J., Hanshar, F.: 'Multi-objective genetic algorithms for vehicle routing problem with time windows', *Appl. Intell.*, 2006, **24**, (1), pp. 17–30
[16] Solomon, M.: 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Oper. Res.*, 1987, **35**, (2), pp. 254–265
[17] Sicilia, J.A., Quemada, C., Royo, B.: 'An optimization algorithm for solving the rich vehicle routing problem based on variable neighborhood search and tabu search metaheuristics', *J. Comput. Appl. Math.*, 2015, **291**, pp. 468–477
[18] Nalepa, J., Blocho, M.: 'Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows', *Soft Comput.*, 2016, **6**, pp. 1–19
[19] He, Y.: 'A tabu search algorithm with variable cluster grouping for multi-depot vehicle routing problem'. IEEE, International Conference on Computer Supported Cooperative Work in Design IEEE, Hsinchu, Taiwan, 2014, pp. 12–17
[20] Yao, B., Yu, B., Hu, P.: 'An improved particle swarm optimization for carton heterogeneous vehicle routing problem with a collection depot', *Ann. Oper. Res.*, 2016, **242**, (2), pp. 1–18
[21] Lzaqebah, M., Abdullah, S., Jawarneh, S.: 'Modified artificial bee colony for the vehicle routing problems with time windows', *Springerplus.*, 2016, **5**, (1), p. 1298
[22] Chaovalitwongse, W., Kim, D., Pardalos, P.M.: 'GRASP with a new local search scheme for vehicle routing problems with time windows', *J. Comb. Optim.*, 2003, **7**, (2), pp. 179–207
[23] 'Research journals'. Available at http://w.cba.neu.edu/~msolomon/problems.htm, accessed March 2005
[24] 'Research journals'. Available at https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark, accessed April 2008

This is an open access article published by the IET under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by-nc-nd/3.0/)