



Contents lists available at ScienceDirect

Omega

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



A case-based reasoning approach to solve the vehicle routing problem with time windows and drivers' experience[☆]

Olivier Quirion-Blais, Lu Chen^{*}

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:

Received 4 September 2019

Accepted 16 September 2020

Available online xxx

Keywords:

Case base reasoning (CBR)

Drivers' experience

Artificial intelligence

Routing

Optimization

ABSTRACT

In last-mile delivery, on-line retailers deliver goods from local distribution centers to endpoint customers using a fleet of vehicles. This problem is often related to vehicle routing problems with time windows (VRPTWs) in the literature. For an on-line retailer in China, it was found that experienced drivers could often find better routes rather than relying on computerized tools using state-of-the-art algorithms. Therefore, the focus of this paper is to generate routes based on experience. To do so, we propose a methodology based on case base reasoning (CBR). The methodology designs new routes to fulfill orders by retrieving and adapting routes previously performed from a repository named case base. A mechanism is also developed to maintain good quality routes in the case base. The methodology is first tested on problem instances generated using a construction heuristic. Other tests are also performed using real data from an on-line retailer in China. Results show that the CBR methodology designed can effectively generate routes to solve new problems similar to routes previously performed. A comparison to the BoneRoute algorithm show that the solutions obtained with CBR are in average 18.4% longer. However, this result does not take into consideration the time required by the drivers to adapt to a very different route.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Vehicle routing problems (VRPs) consist in finding the optimal set of routes departing from a single depot and visiting a set of customers dispersed in a geographical area. Vehicle routing problems with time windows (VRPTWs) are an extension of VRPs which involve visiting customers within a given time interval and ensuring that the total demand of all customers in each route respect vehicles' capacity. This type of problem is encountered on a daily basis by on-line retailers in the final stage of their distribution networks. Goods are gathered in distribution centers from where truck drivers begin their route to deliver goods to the final recipient.

This work is based on a collaboration with one of the largest on-line retailers in China. During regular operations, the company assigns a subset of customers to each driver. Then the drivers have the freedom to decide the order in which they will visit these customers. To help them in this task, the on-line retailer provides an in-house computerized tool to drivers. However, in urban contexts, there are numerous environmental

conditions that affect the performance of the routes and these are often difficult to model mathematically. For instance, drivers might want to avoid roads prone to congestion, unfamiliar roads or city centers. Other examples also include some traffic lights in pedestrian area which make it more difficult to turn than to go straight. Some narrow streets make it more difficult for delivery vehicles to maneuver. Thus, even if the computerized tool is based on state-of-the-art algorithms, drivers can often find better routes thanks to their experience.

In practice, it means that drivers can find more efficient paths than the theoretical shortest path. In the literature, Bates et al. [3] and Ottomaneli and Wong [49] proved that drivers might prefer not to take the shortest path suggested by state-of-the-art algorithm. Although the drivers' experience seems to be an important factor when designing routes, a recent study by Srivatsa Srinivas and Gajanand [62] establishes that there is no methodology considering it to solve VRPTW.

In this context, this paper aims to develop a methodology that focuses on the utilization of drivers' experience. The methodology is based on case-based reasoning (CBR) which takes its origin from the field of artificial intelligence. The idea is to store past experience in the form of "cases" in a database or "case base". When a new problem is encountered, the most relevant cases are retrieved and used to build a solution similar to past experience.

[☆] Area: Optimization; Last-Mile Delivery and Applications. This manuscript was processed by Associate Editor John Zhu.

^{*} Corresponding author.

E-mail address: chenlu@sjtu.edu.cn (L. Chen).

The rest of this article is organized as follows, related works are presented in Section 2. Then the description of the problem studied in this paper is provided in Section 3. The CBR methodology is developed in Section 4. In Section 5, the performance of the methodology is assessed using generated data and data from an on-line retailer in China. Concluding remarks are given in Section 6.

2. Related works

In the scientific literature, drivers' experience is closely related to drivers' behavior. Therefore, in the following subsections, latest works concerning VRPs, drivers' behavior and CBR are presented. They are followed by the scientific contribution of our paper.

2.1. VRPs extensions and solution approaches

The origin of VRPs can be traced back to the paper by Dantzig and Ramser [14]. The problem they described consists in routing a fleet of vehicles from a bulk terminal to supply a set of service stations with different types of products. Due to their wide range of applications, VRPs quickly attracted a lot of attention. The books by Toth and Vigo [67], [68] and Golden et al. [27] outline the milestones in the development of the problem. Numerous variants have been proposed to adapt VRPs to real life contexts. In a effort to standardize those variants, Eksioglu et al. [17], Lahyani et al. [42] and Braekers et al. [6] introduced a taxonomy.

To handle large quantities of data faced in real life, many heuristic methodologies were put forward to solve VRPs. Traditionally, heuristics have focused on rather specific versions of VRPs. Among all VRP variants, Gayialis et al. [22] found that VRPTWs are often selected to model problems related to urban freight. Multiple literature reviews have been released about VRPTWs such as the ones from Bräysy and Gendreau [8], [9], Gendreau and Tarantilis [23], [2] or Desaulniers et al. [15]. Savelsbergh [58] proved that the VRPTW was NP-hard for a fixed number of vehicles. To address instances encountered in practice, many heuristic methodologies have been proposed [43]. Among others, Mester and Bräysy [48] proposed a metaheuristic combining guided local search and evolution strategies and Hashimoto et al. [29] proposed an iterated local search based on dynamic programming.

Recent research trends have focused on broadening the scope of heuristics by encompassing many attributes, often including time windows. For example, Cordeau and Maischberger [12] combined iterated local search and tabu search to solve four VRPTWs. Another metaheuristic offering good performance is the unified hybrid genetic search metaheuristic proposed by Vidal et al. [70]. Thanks to its component-based structure, this metaheuristic can efficiently solve a broad range of VRPs, including VRPTWs.

2.2. VRPs related to drivers' behavior

The VRP extensions that were found to be the most similar to the problem of finding routes considering drivers' experience are summarized in Table 1. One of the first extensions is the courier delivery problem (CDP) introduced by Sungur et al. [64]. In this problem, the objective is to service returning customers at roughly the same time every day. This is done by building a master schedule and adapting the schedule to the daily variations. Time windows and uncertain service times are also considered when solving the problem.

Another extension, close to the CDP, is the consistent vehicle routing problem (ConVRP). This extension was first introduced by Groër et al. [28]. In contrast to the CDP, the ConVRP does not consider time windows and uncertainty in service times, but it extends the notion of consistency to other aspects as described in

Kovacs et al. [38]. Time consistency consists in visiting the same customers at roughly the same time from one day to another. Delivery consistency consists in delivering roughly the same quantity every time or keeping time intervals between deliveries constant. Person consistency ensures that customers are always serviced by the same drivers, drivers always visit the same customers and drivers are always assigned to the same geographical region. Feillet et al. [18] focused on time consistency and developed the time classes model to measure the degree of consistency in the time-consistent vehicle routing problem (TCVRP). Luo et al. [46] consider driver consistency along with time windows and visiting quotas in the multi-period vehicle routing problem with time windows and limited visiting quota (MVRPTW-LVQ). Kovacs et al. [40] studied the GenConVRP, a generalized version of the ConVRP in which the number of drivers visiting each customers is limited and variation in the arrival time is penalized in the objective function. AM/PM time windows are also enforced. Kovacs et al. [41] considered both driver and time consistency along with time minimization in a multi objective problem, namely the multi-objective generalized consistent vehicle routing problem (MOGenConVRP). Recently, Campelo et al. [10] addressed the ConVRP with service agreements such as time windows and release dates.

An exact solution approach for the ConVRP is proposed by Goeke et al. [24]. Heuristic approaches have also been proposed. Many of them consist in partitioning the customers into two groups: regular and irregular. A master template containing one route per vehicle ensures service to all regular customers. The master template is then updated to accommodate the daily demand. This type of approach has been used by Groër et al. [28], Kovacs et al. [39], Tarantilis et al. [66], among others.

The vehicle routing problem with time windows and driver specific times (VRPTWDST) considered personalisation of driving time. This problem is systematically investigated in Schneider [59]. The author assumes that when drivers are regularly assigned to the same geographical area, they get accustomed to the area. The familiarity of drivers to their sector is modeled using a parameter reducing duration of routes. Each driver has its own set of familiarity parameters for every sector. In the paper by Schneider [59], a mixed integer programming (MIP) formulation is proposed to model the problem. It is then solved using a tabu search algorithm.

When it comes to solving methodologies, the concept of adaptive memory (AM) developed by Rochat and Taillard [56] can also be used to build routes considering drivers' behavior. The idea behind AM is that good solutions to a problem are likely to contain some components appearing frequently in solutions of lower quality for the same problem. Thus, AM generates a routing plan by using components appearing frequently in a pool of previous solutions. The same problem is solved many times and the pool is updated as better solutions are found. In Rochat and Taillard [56], the solution components reused are routes. Based on this concept, Tarantilis and Kiranoudis [65] developed the BoneRoute algorithm which reuses "bones", or sequences of consecutive customers, instead of entire routes. This algorithm can successfully solve the VRP. In Zachariadis et al. [72] the BoneRoute algorithm was upgraded by storing routes in the pool instead of solutions. A mechanism considering previous usage of the bones was also introduced to help extend the search domain of the algorithm. This last algorithm can successfully solve the vehicle routing problem with simultaneous pick-ups and deliveries (VRPSPD).

The problem tackled in this article consists in considering drivers' experience when generating routes. Drivers' experience is obtained from data collected from historical routes. In this case, person consistency is desirable in the sense that a driver regularly assigned to the same geographical area will learn more about his/her region and will produce better routes. Delivery consistency is also desirable as it will help in reusing previous routes if the

Table 1
Some combinatorial optimization problems related to drivers' behavior found in the literature.

Reference	Problem addressed	Type of methodology	Type of instance	Largest size	Objective
Groër et al. [28]	ConVRP	Template-based record-to-record travel algorithm heuristic (ConRTR).	Simulated instances.	1000	Minimize total travel time.
Sungur et al. [64]	CDP	Two-phase heuristic involving tabu search.	Real-life data set.	3715	Multi-objective, maximize the number of customers served, minimize the number of vehicles, the total earliness and lateness.
Kovacs et al. [38]	ConVRP	Template-based adaptive large neighborhood search heuristic.	Simulated instances.	5178	Minimize total travel time.
Tarantilis et al. [66]	ConVRP	Tabu search metaheuristic.	Simulated instances.	1000	Minimize total travel time.
Feillet et al. [18]	TCVRP	Large neighborhood search metaheuristic.	Simulated instances.	199	Bi-objective, minimize total travel cost and number of time classes.
Luo et al. [46]	MVRPTW-LVQ	Three-stage heuristic involving tabu search.	Simulated instances.	12	Bi-objective, minimize the number of vehicles and total travel cost.
Kovacs et al. [41]	GenConVRP	Large neighborhood search metaheuristic.	Simulated instances.	100	Bi-objective, minimize the total travel time and the maximum arrival time difference.
Kovacs et al. [40]	MOGenConVRP	Two exact solution approaches and one multidirectional large neighborhood search metaheuristic.	Simulated instances.	199	Multi-objective, minimize total travel cost, improving driver consistency and arrival time consistency.
Schneider [59]	VRPTWDST	Tabu search metaheuristic.	Simulated instances.	100	Bi-objective, minimize travel distances and working duration.
Goeke et al. [24]	ConVRP	Exact resolution methodology and large neighborhood search metaheuristic.	Simulated instances.	199	Minimize total travel time.
Ulmer et al. [69]	VRPBDCE	Construction heuristic and neighborhood search associated with a Cost Function Approximation (CFA) to favour drivers' learning in early stages.	Simulated instances.	120	Bi-objective, minimize routing and service costs.

same customers order roughly the same amount of goods. However, these two types of consistency are not a requirement. As for time consistency, we do not see direct relationship with drivers' experience as the time of visit of a customer can vary considerably in historical routes.

As for driver specific travel time, drivers' experience is based on the same assumption that drivers' accustomed to a given geographical sector will perform better. However, in the VRPTWDST it is assumed that an experienced driver performs the same path faster. With drivers' experience, the assumption is that drivers will possibly find better paths than those generated by algorithms minimizing travel times. These paths are reused to generate better routes.

The vehicle routing problem with binary driver-customer familiarity (VRPBDCE) is another version of VRPs that considers drivers' experience [69]. In this problem, a driver acquires experience after his first visit to a customer. In subsequent passages of the drivers to the customers he already visited in the past, the service time is considerably reduced. The methodology developed for this problem balances the cost of introducing drivers to new customers and the routing costs in subsequent periods. The idea is to minimize costs over a longer period to time by allowing a restricted set of drivers to visit each customer. However, regularity in the roads taken by the drivers is not specifically taken into account.

Finally, AM provides interesting insights when it comes to reusing previous solutions. However, the methods based on this concept are designed to generate a pool of solutions for a specific problem. In our case, new routing plans are based on historical solutions which usually contain additional customers not included in the problem. Moreover, algorithms like BoneRoute from Tarantilis and Kiranoudis [65] or Zachariadis et al. [72] include a step where local search is performed on the extracted components. This step

is likely to modify the components which is not desirable in our case.

2.3. Statistical and data models of drivers' behavior

In the literature, different methods were used to model drivers' behavior. These models are used to determine the factors influencing the decisions of the drivers and the choices made. For example, Dia [16] used an agent-based approach. Drivers' characteristics were extracted from a survey analyzing their behavior in the context of traffic congestion. This information was used to program agents representing drivers. It was demonstrated that these agents can make realistic choices inside a simulation. In another study, Pahlavani and Delavar [51] used a locally linear neuro-fuzzy model to predict the routes after learning from the experience of the drivers. Prospect theory is also a popular choice when it comes to model the behavior of the drivers (e.g. see Li and Hensher [44]). Even though these methods provide good results, they require to conduct surveys among drivers which can be complicated and time consuming.

Other studies focused on determining the conditions that would influence drivers' decisions. Pang et al. [52] extracted the main factors from surveys and used it to build guidance for a single vehicle. Sadeghi Niaraki and Kim [57] proposed standard information structures based on computer ontology to facilitate the integration of these factors into algorithms.

A study realized by Sun et al. [63] confirms that drivers' decisions are based on more factors than merely travel time and cost. Among others, they pinpoint the following factors: the shipping and driver's employment terms, the entity responsible for toll charges and fuel costs, the shipment content, predictability of travel time, availability of parking to stop on the way and the avail-

ability of fuel stations. Sun et al. [63] also add that at the planning stage, the majority of drivers will use their experience when choosing their path. Srivatsa Srinivas and Gajanand [62] found that driving behaviors are related to the congestion settings. Vehicles that operate in congested areas tend to give more importance to avoiding congestion and saving cost. Vehicles which operate on non-congested networks tend to focus more on the type of road they choose to ensure higher travel speeds. Similar conclusions were obtained by Feng et al. [19], but the correlation is based on the size of the trucks. This categorization can be related to the distribution stages found in the two-echelons problem described by Perboli et al. [53].

2.4. CBR applications

One of the first CBR algorithm implementation was named CYRUS. It was proposed by Kolodner [36], [37]. The algorithm solves problems by seeking and adapting experiences, or cases, that are stored inside a case base. The algorithm also makes sure that only high-quality experience is maintained in the case base by implementing a case base management system. For in-depth explanations about the algorithm the reader can refer to the paper from Aamodt and Plaza [1] and the book from Richter and Weber [54].

Goel and Diaz-Agudo [26] identified different usages of CBR in domains such as recommender systems [7], conception and design [25], education [35], law [55], knowledge discovery and interactive robotics [20,21,31] as well as text recognition [61,71]. Chakraborty and Boral [11] provide a good description of the implementation of a CBR algorithm applied to machine tool selection.

There are also a few attempts to solve combinatorial optimization problems. Beddoe et al. [4] used CBR to solve the nurse rostering problem. In their study, the algorithm was used to help health care managers to modify work schedules of nurses to reduce the number of constraint violations as much as possible. The cases they designed contained steps to fix some given types of violation. Since the cases intend to fix schedule violations one at the time, the CBR algorithm was implemented along with a memetic algorithm which selects the order in which the violations should be addressed.

In the field of transportation, Bouhana et al. [5] implemented CBR along with Choquet integral to generate vehicle guidance for people displacement in a multi-modal context. It was also used to design guidance for a road vehicle under different circumstances. For example, Haigh et al. [30] exploited geographical properties to retrieve one or multiple routes that were used in the past and merge them into one final route for the user. McGinty and Smyth [47] also used CBR for vehicle guidance, but in their case, they developed a framework allowing one user from a geographical region to use the case base of similar users from another geographical region. Then a driver can get a personalized route in a region where he has never been before. More recently, Sedano et al. [60] developed a CBR methodology for the purpose of guiding electric vehicles to the nearest charging station.

CBR was also used to solve the more general Traveling salesman problem (TSP). In their study, Louis and Li [45] used the retrieved cases to improve the quality of the initial population of a genetic algorithm. It was shown that the use of CBR led to an increased performance. Kawabe et al. [34] also obtained solutions for the TSP in a very short time with only a slight decrease in the quality using CBR.

2.5. Contribution of this work

When it comes to selecting a path to reach customers, some studies stress the importance of considering factors other than cost and time when planning paths for drivers. In the context of VRPs,

some papers do consider the quality of service by integrating service consistency. Other papers consider familiarization of drivers in geographical zone in which they work frequently. As far as we know, there is no contribution in the literature that considers drivers' experience in the sense of sequence of customers to be followed in the context of VRPs. In that sense, the main contributions of this work are:

1. Development of a methodology based on CBR to maintain high quality experience to build solutions based on past experience. According to Cunningham [13], CBR has the capacity to provide good solutions in situations where features affecting solutions are difficult to formalize.
2. Formalization and storage of drivers' experience in a case base. Similarity criteria and MIP formulation are developed to retrieve routes from the case base to ensure an efficient utilization of past experience.
3. Illustration of the effectiveness and the efficiency of our approach using instances generated from a real distribution network, as well as instances with real routes. The feasibility of applying our approach in the real practice is also analyzed.

3. Problem specification

The problem addressed in this paper is encountered on a daily basis by an on-line retailer for last-mile delivery, when goods are being delivered from local transfer stations to endpoint customers. Each transfer station employs a limited number of drivers. When they are planning deliveries, logistic managers evaluate the workload and assign orders using a computerized tool. The same tool is also offered to the drivers responsible for delivery. However, experienced drivers can find paths that are more efficient than those provided by the computerized tool. Incorporating drivers' experience would improve the efficiency of the decision-making process when designing routes for each driver. Since drivers' experience take the form of routes performed previously, the problem consists in finding the routes that are the most similar to a new problem instance encountered. Thus, the decision-making process does not rely on drivers who have different levels of experience.

The concept of similarity used in this work is based on **customer chains** that we define as sequences of consecutive customers found in a route. When it comes to considering customer chains, two characteristics are important, the number of customer chains and their length. Therefore, to be similar to historical data, a given solution should contain as many lengthy customer chains as possible. To consider historical data, we introduce Ω , the set containing all customer chains that were performed by the drivers in the past.

We now present a MIP formulation for the VRPTW with drivers' experience. Table 2 summarizes the notation used in the formulation. The problem can be modeled using a set partitioning MIP formulation [15]. Let C be the set of all customers requiring service. F is the set of routes starting at the depot, visiting a subset of C and ending at the depot. Let Ω_j be the set of all customer chains performed historically found in route $j \in C$. p_j is the cost of using route j . Let a_{ij} represent if customer i is visited in route j and b_k be the square value of the number of customers in chain $k \in \Omega_j$. Variable z_i indicates whether route $i \in F$ is selected by the formulation or not.

The master problem is presented in Eqs. (1) to (4). The solution consists in finding a subset of routes from F , the set of all feasible routes, that visit each of the nodes exactly once and which respect the time windows and the capacity constraints.

$$\text{maximize} \quad \sum_{j \in F} z_j \sum_{k \in \Omega_j} b_k \tag{1}$$

Table 2
Notation used in the MIP formulation of VRPTWs.

Sets	
C	customers required in the new problem, $C = \{c_1, c_2 \dots c_{ C }\}$, indexed by i ;
F	set of all feasible routes $F = \{r_1, r_2 \dots r_{ R }\}$, indexed by j .
Ω	all customer chains obtained from historical data;
Ω_j	customer chains of variable length in route j in the historical data or in the final solution, indexed by k ;
Parameters	
a_{ij}	=1 if customer i is visited in route j ; 0 otherwise;
b_k	the square value of the number of customers in chain k ;
p_j	total cost of route j .
Variables	
z_j	=1, if route j is selected by the master problem or if the CBR methodology retrieves the route; 0, otherwise.

$$\text{minimize} \quad \sum_{j \in F} p_j z_j \tag{2}$$

$$\text{s.t.} \quad \sum_{j \in F} a_{ij} z_j = 1 \quad \forall i \in C, \tag{3}$$

$$z_j \in \{0, 1\} \quad \forall j \in F. \tag{4}$$

To consider drivers’ experience, we propose to use a multiobjective formulation where Eq. (1) maximizes the number of lengthy historical customer chains in the solution and Eq. (2) minimizes the total cost incurred in the solution. To account for the length of customer chains in Eq. (1), parameters b_k are the square values of the number of customers found in the chain. The square values are used to increase the emphasis on longer chains which are considered as more similar. For instance, one chain of four customers is deemed more similar than two sequences of two customers. The total cost in the second objective function (2) accounts for three types of cost:

1. fixed cost for each vehicle used: for this problem, it is considered that drivers are given a fixed amount for each working day. Therefore the fixed cost mainly accounts for the salary of the employees. It also accounts for fixed charges related to vehicles.
2. variable cost for distance travelled: this part of the cost accounts for the wear and tear of the vehicles.
3. variable cost for waiting time: even if waiting times are allowed theoretically, employees will often overlook waiting times, resulting in time windows not being respected. Penalizing waiting times reduce these kinds of situation.

Constraints (3) ensure that each customer is visited exactly once. Constraints (4) define the variable in the master problem.

To generate the routes in F , column generation algorithms were proposed, for example the reader can refer to Irnich and Desaulniers [33]. In our case, the constraints considered in the subproblem are the time windows and the vehicle capacities, both in terms of volume and weight.

To simplify the problem, it is common to cluster together individual civic addresses that are in close proximity. These clusters are the nodes in the network. For the rest of the article, we consider that one customer is one cluster containing many civic addresses.

4. A CBR methodology to solve the VRPTW

Before detailing the methodology proposed for the problem, we define the following terms. **Required customers** are the customers that need to be serviced in the new problem. **Retrieved routes** are the routes from the case base that are the most similar to the new problem. As there might be **excess customers** in the routes, i.e.

customers that appear more than once in the retrieved routes, we define **selected customers** as the customers that will be kept for the final solution. Similarly to selected customers, **selected customer chains** are customer chains that will be kept for the final solution. Additional notation for the methodology is also presented in Table 3.

The core idea of CBR is to store past experience obtained from historical data and reuse it to design future solutions. For the suggested methodology, the experience stored are routes previously performed by drivers. Each route is considered as a case stored in the case base. Given a new VRPTW to solve and a case base, the final solution consists in a set of routes visiting all required customers.

It is interesting to denote the similarity between the set partitioning formulation and the CBR methodology. Both methods consist in selecting some routes among a larger set of routes. However, in the set partitioning formulation, the routes selected by the formulation must match exactly the set of required customers and minimize costs. In the CBR methodology, the routes in the case base might contain some excess or missing customers and the time windows are not necessarily respected. Therefore, some other mechanisms must cope with this. Moreover, the objective of CBR is to maximize similarity between the retrieved routes and the new problem. A MIP formulation is used to retrieve routes from the case base.

The methodology is divided into four main steps also called the four Re’s. These steps are depicted in Fig. 1. The objective of the *Retrieval* step is to find the routes in the case base that are the most similar to the new problem. In the *Reuse* step, excess customers are removed from the retrieved cases. Missing customers, i.e. customers that are required in the new problem, but not present in the retrieved cases are also inserted at this step. The objective of the *Revise* step is to adjust the provisional solution to ensure that capacities are respected. Finally, in the *Retain* step, the case base is updated as required.

In the following subsections each step of the methodology is detailed. Then an application example is presented.

4.1. Retrieval

The *Retrieval* step consists in retrieving the most similar routes from the case base. The similarity criterion is based on the number of customer chains that are both present int the set of required customers (C) and the set of cases retrieved (B_{ret}).

A MIP formulation is used to retrieve the routes and select the customers to maximize the similarity. The model is defined in Eqs. (5) to (14).

$$\text{maximize} \quad s = o_1 s_1 - o_2 s_2 + o_3 s_3 - o_4 s_4 \tag{5}$$

Table 3
Additional notation used for the methodology.

Sets	
ω_j	customer chains of fixed length set by the user in route j , indexed by k ;
v_j	customers in route j , indexed by i ;
v_k	customers in customer chain k , indexed by i ;
B	routes in the case base, $B = \{r_1, r_2 \dots r_{ B }\}$, indexed by j ;
B_C	subset of B containing all routes having at least one customer chain, i.e. $B_C = \{j \in B \mid \omega_j \geq 1\}$, indexed by j ;
B_{ret}	subset of B containing all routes retrieved;
R_{prov}	routes in the provisional solution;
R_v	routes requiring revision.
Parameters	
d_{ij}	travel time from customer $i - 1$ to customer i , where $i \in v_j$ (service time is included in the travel time);
e_i	earliest arrival time at customer $i \in C$;
f_i	latest arrival time at customer $i \in C$;
l	number of customers in the customer chains considered by the methodology;
n	size of the case base;
o_1	weight affecting the number of customer chains selected;
o_2	weight affecting the number of routes retrieved;
o_3	weight affecting the number of customers selected;
o_4	weight affecting the total number of customers in the retrieved routes;
w	weight used in the case base management system.
Variables	
t_{ij}	time at which customer i is visited in route j ;
y_{kj}	=1, if customer chain k in route j is selected by the model; 0, otherwise;
x_{ij}	=1, if customer i in route j is selected by the model; 0, otherwise.

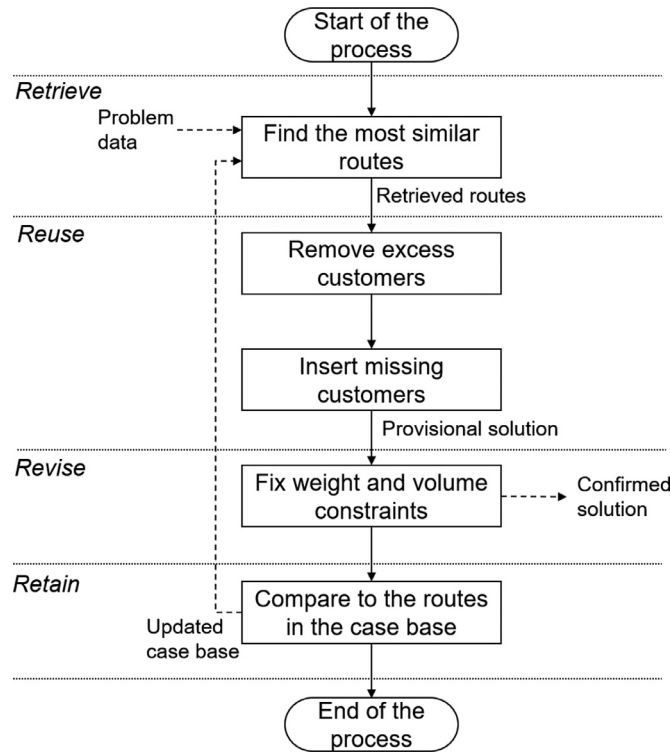


Fig. 1. Framework of the CBR methodology.

$$\sum_{j \in B_C} x_{ij} \leq 1 \quad \forall i \in C \quad (9)$$

$$t_{i-1,j} + d_{ij} \leq t_{ij} \quad \forall i \in v_j, j \in B_C \quad (10)$$

$$t_{ij} \geq x_{ij}e_i \quad \forall i \in C \cap v_j, j \in B_C \quad (11)$$

$$t_{ij} \leq (1 - x_{ij})M + f_i \quad \forall i \in C \cap v_j, j \in B_C \quad (12)$$

$$x_{ij}, y_{kj}, z_j \in \{0, 1\} \quad \forall i \in C \cap v_j, j \in B_C, k \in \omega_j \quad (13)$$

$$t_{ij} \in \mathbb{N}^0 \quad \forall i \in B_C \quad (14)$$

The objective function (5) maximizes the similarity function s . s can be divided into four terms detailed in Eqs. (15) to (18). They are presented in order of priority: s_1 is more important to optimize than s_2 , s_2 is more important to optimize than s_3 and so on. The negative sign before s_2 and s_4 ensures that these terms are minimized.

s_1 measures the number of selected customer chains. s_2 measures the number of routes retrieved. It is affected by a negative sign to minimize it. If two customer chains can be selected either in one route or in two routes, only one route should be selected as it potentially reduces the number of vehicles required in the solution. s_3 measures the total number of selected customers. Even though standalone customers are less interesting than customer chains, it is still useful to maximize them. With the lowest priority, s_4 measures the number of customers in the retrieved routes. This part of the objective ensures that between two routes servicing the same customers, the one having less customers in total is selected. This term also bears a negative sign to minimize its value. To ensure that priorities between the terms are respected, the parameters o_1 , o_2 , o_3 and o_4 affect respectively s_1 , s_2 , s_3 and s_4 . Values for these parameters are suggested in Section 5.2.

$$s_1 = \sum_{j \in B_C} \sum_{k \in \omega_j} y_{kj} \quad (15)$$

$$s.t. \quad y_{kj} \leq \frac{\sum_{i \in v_k} x_{ij}}{l} \quad \forall k \in \omega_j, j \in B_C \quad (6)$$

$$y_{kj} \leq z_j \quad \forall k \in \omega_j, j \in B_C \quad (7)$$

$$\sum_{k \in \omega_j} y_{kj} \geq x_{ij} \quad \forall i \in C \cap v_j, j \in B_C \quad (8)$$

$$s_2 = \sum_{j \in B_c} z_j \tag{16}$$

$$s_3 = \sum_{j \in B_c} \sum_{i \in C \cap v_j} x_{ij} \tag{17}$$

$$s_4 = \sum_{j \in B_c} |v_j| \cdot z_j \tag{18}$$

In constraints (6), a parameter l is introduced to customize the length of the customer chains. This set of constraints thus ensures that customer chain k in route j is selected only if the l x_{ij} 's corresponding to that chain are selected. Constraints (7) impose that the value of z_j is greater than any of the y_{kj} in the same route. It means that if the model selects a customer chain in a route, then the model must also retrieve the route. Constraints (8) allow the selection of x_{ij} 's in route j only if there is at least one customer chain y_{kj} selected. Constraints (9) allow only one x_{ij} to be selected by the model for each customer i in the problem. Constraints (10) to constraints (12) are optional and they ensure the time windows are respected. In details, constraints (10) take into account the time to travel from one customer to another in the route as it is in the case base. Constraints (11) ensure that the customers that are selected by the model are visited after the earliest time allowed. Similarly, constraints (12) ensure that the customers that are selected by the model are visited before the latest time allowed. Finally, constraints (13) and (14) are used for the definition of the variables. It should be noted that up to this point, capacities of the vehicles are intentionally omitted. They will be considered in the *Revise* step.

4.2. Reuse

There are two phases in the *Reuse* step: removing excess customers from the retrieved routes and inserting the missing customers. The customers removed from the routes are those that are not selected by the model. To insert customers, a lowest insertion cost algorithm is used. It consists in iteratively inserting the customer that has the lowest insertion cost among all customers that need to be inserted. At this point of the methodology, capacity and time windows constraints enforced. Therefore, it is possible that some customers cannot be inserted if no insertion point respecting the constraints can be found. For these customers, new routes are created. Theses routes are generated using a construction heuristic. To mimic real operations as in practice, the *Impact* heuristic designed by Ioannou et al. [32] is used. In this heuristic, customers are inserted iteratively by selecting the customer that has a combination of the lowest increase in the distance traveled after insertion, the lowest waiting time and the marginal cost on the insertion cost of the other customers that are waiting to be inserted.

At the end of this step, we obtain R_{prov} , a set containing all the routes of the provisional solution.

4.3. Revise

The next phase of the CBR, the revision phase, is intended to fix vehicle capacity violations. Algorithm 1 summarizes the violation fixing process. In the algorithm, the first for-loop verifies if the volume capacity or the weight capacity of each vehicle is exceeded. If so, the routes are inserted in R_v , the set of routes requiring revision. Other routes of R_{prov} are inserted in a new set containing the final solution $R = R_{prov} \setminus R_v$. Then, the routes of R_v are sorted so that the largest violation is selected first. To fix violations, the `Find_lowest_impact_and_insert( $j$ ,  $R$ )` function searches for the customer from j that has the lowest *Impact* score

Algorithm 1 Algorithm to fix weight and volume capacity violation.

```

1: for all  $k$  in  $R_{reu}$  do
2:    $h_k \leftarrow \max(v_k/V, w_k/W)$ 
3:   if  $h_k > 1$  then
4:     Insert  $k$  in  $R_v$ 
5:     Remove  $k$  from  $R$ 
6:   end if
7: end for
8: Sort( $R_v$ )
9: while  $R_v$  is not  $\emptyset$  do
10:   $k \leftarrow S_v(0)$ 
11:  if Find_lowest_impact_and_insert( $k$ ,  $R$ ) then
12:    if respect_all_constraints( $k$ ) then
13:      Insert  $k$  in  $R$ 
14:      Remove  $k$  from  $R_v$ 
15:    end if
16:  else
17:    Add empty route in  $R$ 
18:  end if
19: end while

```

when being inserted in any other routes from R without violating any constraints. If such customer is found, it is transferred, and the function returns true. After the relocation, if route j has no more violations, then it is transferred to R . In the case where `Find_lowest_impact_and_insert( $j$ ,  $R$ )` would return false, an empty route is added to R .

4.4. Retain

The *Retain* step consists in inserting the routes from R in the case base according to certain criteria and insuring a good management of the case base. All routes from R that have more than two customers and that are not duplicate in the case base are inserted in the case base. To ensure good performance of the methodology and to keep the mathematical formulation from the *Retrieval* step tractable, the number of cases stored is limited to n . When the number of routes is greater than n , the routes are sorted using a ranking function h_j given by Eq. (19) and the routes having the highest values are removed. h_j is composed of two terms. h_{j1} accounts for the efficiency of the routes where p_j is the cost of the route. h_{j2} accounts for the diversity in the case base where m_i is the number of occurrence of customer i in the case base. A factor w is also introduced to balance the weight given to each term.

$$h_j = h_{j1} + w \cdot h_{j2} = \frac{p_j}{|v_j|} + w \cdot \left(\sum_{i \in v_j} \frac{1}{m_i} \right)^{-1} \tag{19}$$

4.5. An application example

To illustrate the methodology, an application example of the CBR methodology is presented. The data related to the example are presented in Table 4. The following notation is used: C_{ret} is the list of customers that need to be inserted, updated after the *Retrieval* step. Similarly, R_{ret} is the partial solution obtained after the *Retrieval* step.

In the *Retrieval* step, routes r_1 and r_3 are retrieved. r_2 , r_4 and r_5 are not retrieved for the following reasons: r_2 contains only one required customer, r_4 does not contain any customer chain and r_5 contains a chain of two customers, however the same customers can be found in r_3 , which has a longer chain. Thus, after retrieving r_1 and r_3 , the only customer left for insertion for the subsequent

Table 4
Sample case used to illustrate the methodology.

Initial data	
$C =$	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$
$B =$	$\{r_1: [c_4 \rightarrow c_5],$ $r_2: [c_4 \rightarrow c_{12} \rightarrow c_{13}],$ $r_3: [c_1 \rightarrow c_9 \rightarrow c_2 \rightarrow c_6 \rightarrow c_7],$ $r_4: [c_5 \rightarrow c_{11} \rightarrow c_{10} \rightarrow c_2],$ $r_5: [c_1 \rightarrow c_6 \rightarrow c_{10}]\}$
Retrieval	
$C_{ret} =$	$\{c_3\}$
$B_{ret} =$	$\{r_1: [c_4 \rightarrow c_5],$ $r_3: [c_1 \rightarrow c_9 \rightarrow c_2 \rightarrow c_6 \rightarrow c_7]\}$
Reuse	
$C_{prov} =$	$\emptyset$
$R_{prov} =$	$\{r_1: [c_4 \rightarrow c_3 \rightarrow c_5],$ $r_3: [c_1 \rightarrow c_2 \rightarrow c_6 \rightarrow c_7]\}$
Revise	
$C_{rev} =$	$\emptyset$
$R =$	$\{r_1: [c_4 \rightarrow c_3 \rightarrow c_5],$ $r_3: [c_1 \rightarrow c_2 \rightarrow c_6 \rightarrow c_7]\}$

steps is c_3 . In the *Reuse* step, c_9 is an excess customer and it is removed. On other hand, c_3 is inserted in r_1 considering that time windows and capacities are allowing. Assuming that capacities of the vehicles are respected, no additional changes are performed in the *Revise* step.

5. Performance analysis

In this section, we first calibrate the parameters o_1 , o_2 , o_3 and o_4 . We also look at the performance of the *Retrieval* MIP formulation for problem instances of larger size. Then we use random generated instances to test the performance of the CBR methodology. We look at two different parameters which are: l , the number of customers in the customer chains; and n , the size of the case base. The value of w in the retain mechanism is kept constant. Due to the similarity in their logic, a comparison is made between the results obtained by CBR and BoneRoute. Finally the method is applied using real data from the company.

5.1. Background of the problem

Problem instances are generated using a list of customers coming from the on-line retailer. Two types of instances are generated, smaller instances composed of 25 to 40 customers picked randomly from a list of 100 customers and larger instances composed of 95 to 105 customers picked randomly from a list of 250 customers. For both type of instances, each of the customers has a given time window. The weights and the volumes of the goods ordered by the customers are picked randomly from a list of standard weights and volumes.

To populate the case base during the experiments, we first generate a set of nearly 310000 routes using the *Impact* construction heuristic [32]. A certain number of these routes are then selected following the ranking system explained in Subsection 4.4 to initialize the case base.

The objective of this study is to build routes based on past experience. Therefore, the cost of the new routes is the secondary consideration, especially when the cost of the routes cannot be simply calculated by the shortest path matrix. The quality of the solutions is rather based on the similarity with experience. To measure similarity, we use two metrics: θ_1 and θ_2 . θ_1 is based on the amount of customer chains and the number of customers in the chain. Θ_1 is obtained using Eq. (20). In this equation, $|\omega_k|$ accounts for the number of customers in customer chain k . Values of θ_1 vary between 0% and 100% and values approaching 100% in-

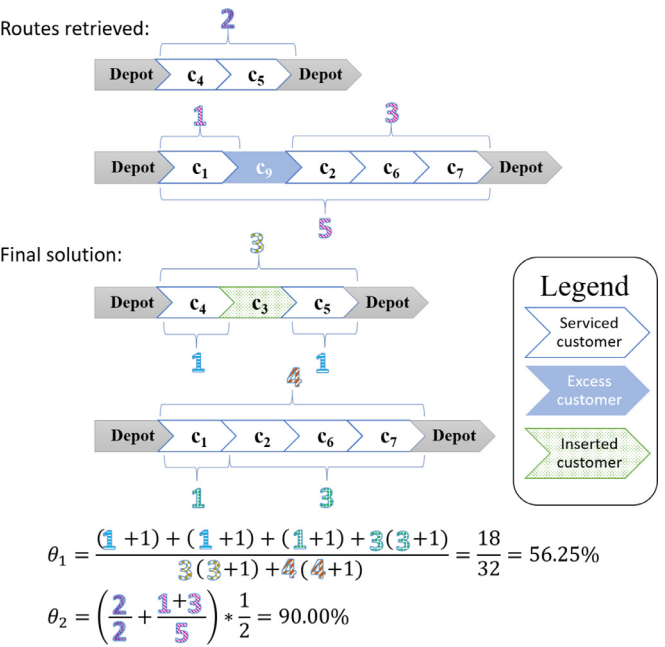


Fig. 2. Example of calculation of θ_1 and θ_2 .

Table 5
Various values for the objective function parameters of the *Retrieval* step. ($n \in \{40000 - 50000\}$, $l \in \{4 - 5\}$).

o_1	o_2	o_3	o_4	θ_1	θ_2
2000	100	50	1	65.48%	54.86%
2000	100	500	1	67.59%	54.16%
2000	0	100	0	65.44%	54.54%
2000	0	100	1	65.58%	54.43%
2000	0	2000	50	65.93%	54.70%
20	0	1000	1	69.47%	52.56%
200	0	1000	1	68.87%	52.51%
1000	0	2000	1	70.37%	52.56%
2000	0	2000	1	70.11%	52.15%
2000	0	2000	500	43.70%	54.34%

dicate a higher degree of similarity between the solutions and the retrieved routes.

$$\theta_1 = \frac{\sum_{j \in B_{ret}} \sum_{k \in \Omega_j} |\omega_k| (|\omega_k| + 1)}{\sum_{j \in R} |v_j| (|v_j| + 1)} \quad (20)$$

θ_2 measures the extent to which the retrieved routes are reused. Its value is given by the percentage of selected customers in the retrieved routes from the case base. Since there is often more than one case retrieved from the case base, we take the average of all the percentages of customers reused. Values of θ_2 vary between 0% and 100% and values approaching 100% indicate a higher degree of reuse of the retrieved routes. Fig. 2 displays an example of the calculation of θ_1 and θ_2 for the sample case in Table 4.

5.2. Calibration of the parameters

The values of o_1 , o_2 , o_3 and o_4 are set according to the experimental results. The objective is to maximize the value of both of θ_1 and θ_2 .

Coarse results in Table 5 allow to select the following values for the parameters : $o_1 = 2000$, $o_2 = 100$, $o_3 = 500$ and $o_4 = 1$. Then, Figs. 3 to 6 show that the results obtained within a limited range of the parameters are fairly stable. All results were obtained using the smaller generated instances.

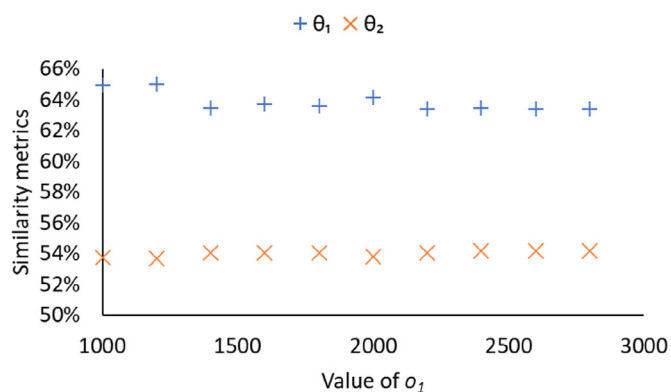


Fig. 3. Sensitivity analysis of o_1 .

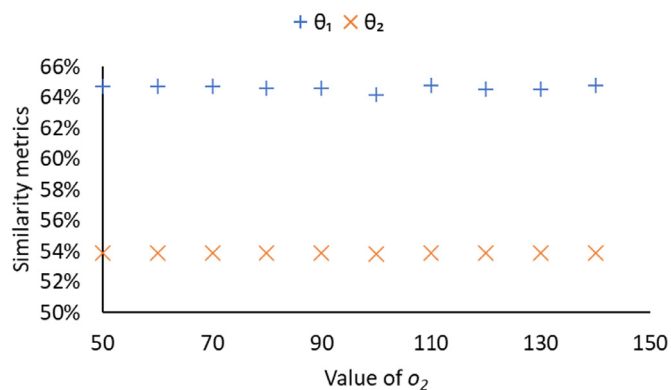


Fig. 4. Sensitivity analysis of o_2 .

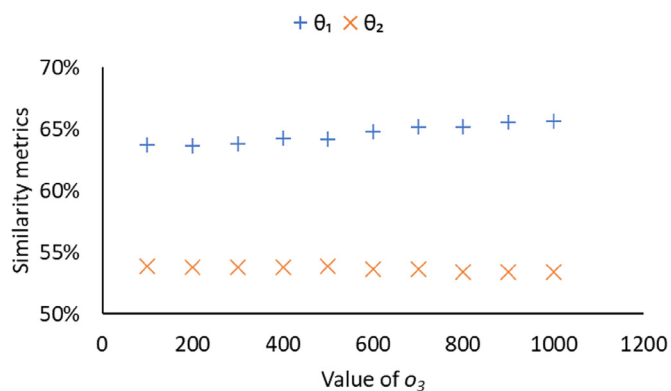


Fig. 5. Sensitivity analysis of o_3 .

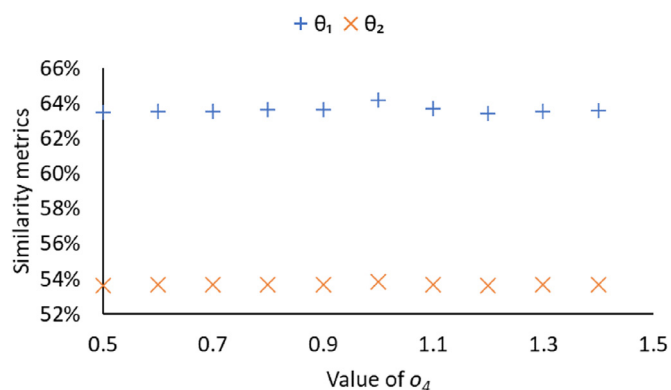


Fig. 6. Sensitivity analysis of o_4 .

Table 6
Average run times over 90 instances (Values averaged for $n \in [10000 - 60000]$).

l	Retrieval run time (seconds)	Time to generate one solution (seconds)
4	505.5033	530.4487
5	64.9764	71.2319
6	0.1080	2.4749
7	0.0092	1.5136

Table 7
Status of the resolution (Values averaged for $n \in [10,000 - 60,000]$).

l	Solved to optimality	Feasible solutions
4	68	22
5	88	2
6	90	0
7	90	0

5.3. Performance of the retrieval MIP formulation

In this section, the performance of the method in terms of run time and the performance of the MIP formulation are studied. For these experiments, we use the larger instances which correspond approximately to the largest instance size that was obtained from the on-line retailer. Among the four steps, the *Retrieve* step, the *Retain* step and the *Reuse* step can require more time. The duration of the *Reuse* step increases linearly with the number of cases retrieved. When many cases are retrieved, the number of insertion positions can be high. However, this part of the algorithm is relatively short in comparison to the *Retrieval* time. The *Retrieval* time requires more details and is discussed in the next paragraph. The duration of the *Retain* step grows exponentially with the number of cases in the case base since the occurrences of the customers need to be updated every time a case is inserted. In our implementations this step did not require more than a few minutes. Moreover, since the *Retain* step can be performed at periods during which computational resources are less busy in the company, this step is not considered in the analysis.

The main operation affecting the duration of the *Retrieval* step is the resolution of the MIP in Eqs. (5) to (14). It is confirmed in Table 6 where the two columns compare the time required to solve the model and the time required for the entire algorithm to generate a solution after the *Revise* step. The MIP formulation is solved by the commercial solver CPLEX 12.8.0. The running time of the solver is limited to 600 seconds. When optimality is not reached, the average gap to the lower bound obtained is 9.20%. Table 7 shows that optimality is more difficult to obtain with lower values of l due to the greater number of variables in the problem. This trend also appears in Table 6 where the average computational time for lower values of l are considerably larger. The overhead added by the rest of the algorithm is only a few seconds which is reasonable.

It could be expected that as the number of routes in the case base increases, the run time would also increase. However, with up to 60000 routes, only a slight increase in the run time was observed.

5.4. Performance analysis of the CBR approach on generated instances

The smaller size instances are used for this part of the analysis. The size of these instances are comparable to the average instance

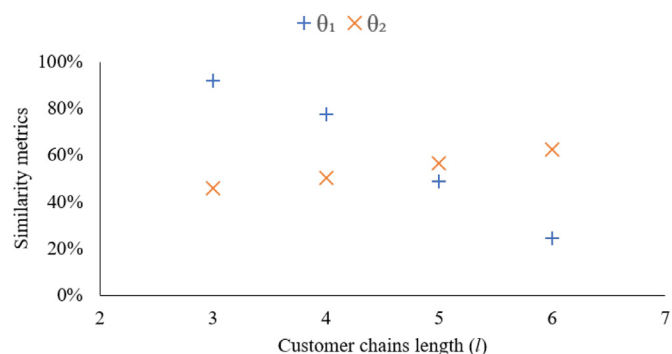


Fig. 7. Effect of the length of the customer chains, l on the percentage of customers retrieved and the route usage (Values averaged from 100 instances. $n \in \{40000, 50000\}$).

Table 8
Statistics about the source cases and various case base sizes.

	μ_c	μ_q
All cases	515.75	19664.15
10000	407.54	719.93
20000	417.36	1620.71
30000	423.64	2398.63
40000	429.59	3158.19
50000	435.18	3900.20
60000	440.51	4626.63

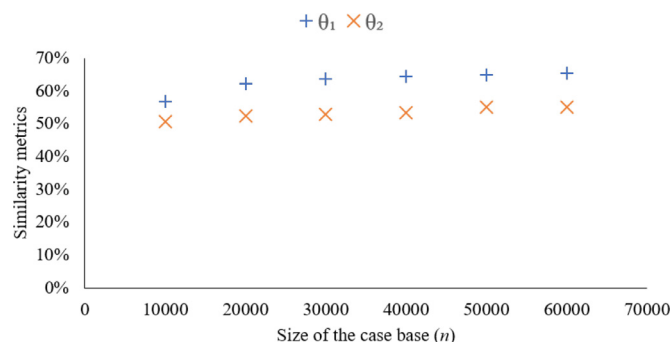


Fig. 8. Effect of n , the size of the case base on θ_1 and θ_2 (Values averaged from 100 instances. $l \in \{4, 5\}$).

size from the on-line retailer. Fig. 7 shows that as the length of customer chains (l) increases, the value of θ_1 decreases and the value of θ_2 increases. This is due to the fact that a higher value of l induces longer customer chains which are more difficult to find in the case base, but increases the percentage of reuse of the retrieved routes. Therefore l can be used to customize the definition of similarity according to the preference of the user. With the current results, values of $l \in \{4, 5\}$ achieve the highest combination of values for θ_1 and θ_2 .

We denote μ_q as the arithmetic mean of the number of occurrences of the customers in the case base and μ_c as the arithmetic mean of the cost of all routes in the case base. μ_q is used to measure the diversity of the case base and μ_c is used to measure the efficiency of the routes in the case base. Results in Table 8 show that as n increases, the diversity increases as well. At the same time, quality decreases since μ_c increases.

Fig. 8 shows that as the size of the case base (n) increases, both θ_1 and θ_2 increase as well. Thus the additional cases inserted in the case base allow to increase the quality of the routes generated. With these results, we can say that when $n = 40000$ or $n = 50000$,

Table 9
Comparison of solution costs $n \in [40,000 - 50,000]$.

ID	Cost Impact	Cost CBR		% Gap	
		$l = 4$	$l = 5$	$l = 4$	$l = 5$
1	15708.20	13421.96	14023.02	15%	11%
2	29065.06	26124.98	16781.12	10%	8%
3	13761.04	12420.71	13189.47	10%	4%
4	16165.71	15049.13	15223.34	7%	6%
5	15080.20	14044.50	13718.00	7%	9%
6	27273.16	25498.03	30613.74	7%	-12%
7	15825.31	14800.50	15795.72	6%	0%
8	27201.90	25786.58	26436.79	5%	3%
9	29311.98	28012.56	26737.43	4%	9%
10	13573.19	12987.36	12862.95	4%	5%
11	12800.68	12634.59	13158.58	1%	-3%
12	13505.56	13512.26	13331.68	0%	1%
13	15224.70	15277.20	14286.32	0%	6%
14	13665.06	13720.03	13124.88	0%	4%
15	26640.33	26917.61	26785.01	-1%	-1%
16	26819.54	27168.83	25784.58	-1%	4%
17	14159.64	14765.29	15223.34	-4%	-8%
18	10830.12	11388.37	11231.24	-5%	-4%
19	13100.12	13810.03	12808.10	-5%	2%
20	24231.17	25559.86	26264.02	-5%	-8%
21	12004.40	12737.89	12439.65	-6%	-4%
22	11003.40	11852.81	11567.98	-8%	-5%
23	24500.85	26476.29	23469.44	-8%	4%
24	25567.38	27696.09	25171.66	-8%	2%
25	11717.95	13345.45	12692.89	-14%	-8%

both θ_1 and θ_2 have reached values close to their maximum values.

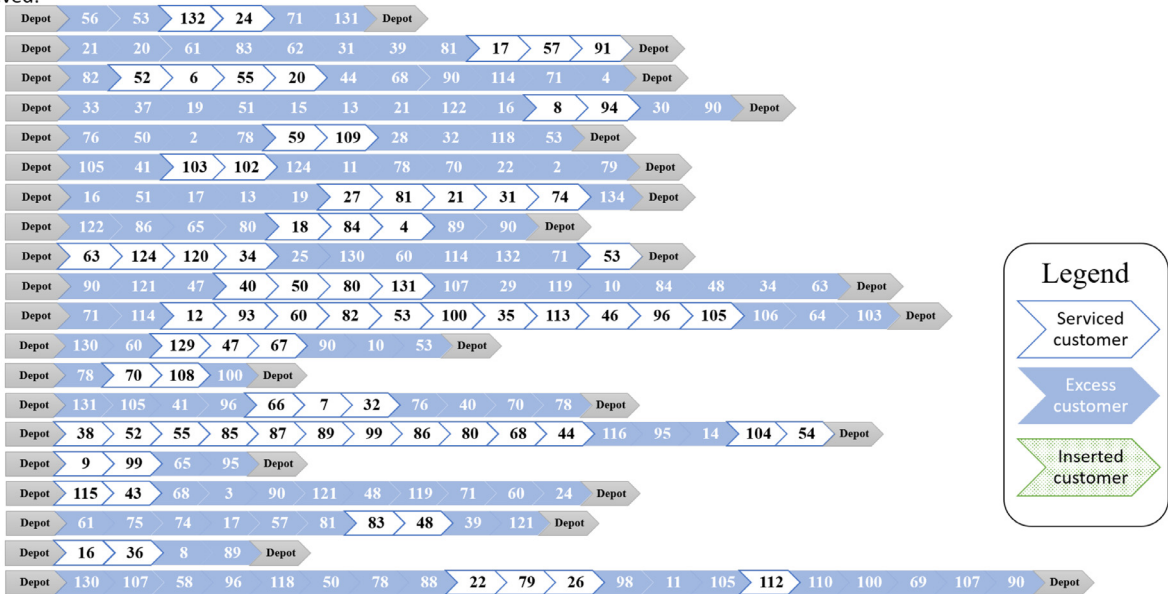
We also look at the impact of the CBR method on the cost. To do so, small size instances are first solved using the *Impact* algorithm. The same problem instances are also solved using the CBR method with the best parameters. Then the results for both methods are compared. Since the solutions obtained using CBR are based on historical solutions generated using *Impact*, it is expected that the costs should be in the same range. Results are presented in Table 9 where the gap is obtained using the following formula: $(Cost_Impact - Cost_CBR)/Cost_Impact$. They show that the cost gap varies from -14% to 15% and the average gap is 0.7%. This range is considered acceptable especially since travel time evaluation already includes some deviation due to various factors such as narrow streets or congestion prone areas.

With the parameter values proposed in the current subsection, average values of $\theta_1 = 67.29\%$ and $\theta_2 = 54.30\%$ were obtained. These values show that in general, the CBR methodology produces solutions that are similar to those in the case base. Since the routes in the case base are based on routes performed in the past, we can assess that the methodology succeeds in generating new routes inspired by driver's experience.

In summary, results show that routes similar to the new problem could be retrieved from the case base. A proper value of l allows to obtain a trade-off between the total number of customers selected by the model and the percentage of customers selected in the retrieved routes. It was also found that the similarity slowly increases with n until a plateau is reached. The value of n should be selected at the point where the curve flattens. A higher value would increase the number of lower quality routes. From the point of view of a retailer, the values of l and n can be used to impose a certain level of similarity of the routing plan. The rest of routes being generated using a state-of-the-art algorithm, we can assume that the costs of the rest of the routes will be lower. Therefore, the value of n and l can also be used as leverage to control the cost and the similarity of the solutions.

To further assess the performance of the CBR methodology, a comparison is made between the results obtained by the CBR

Routes retrieved:



Final solution:

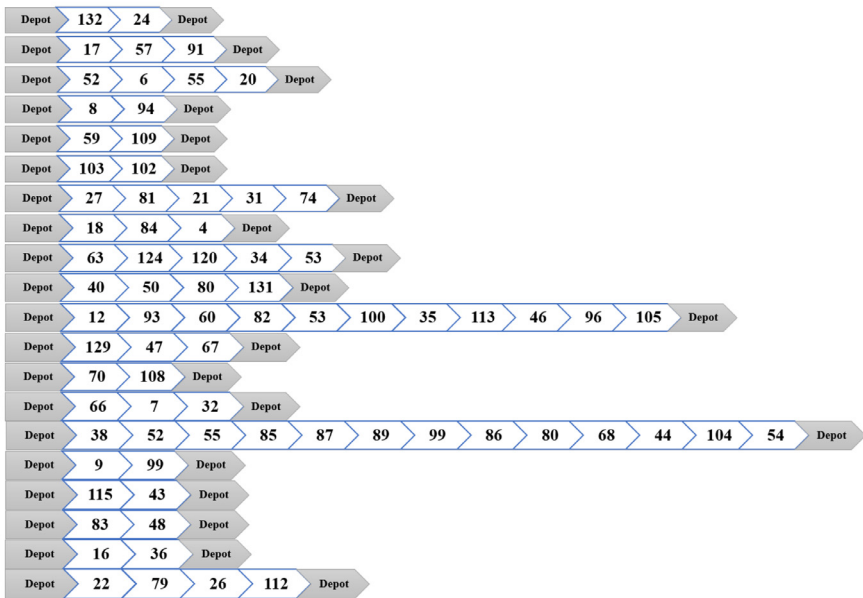


Fig. 9. Comparison of the retrieved routes and the final solution for the real instance *R/3* when using $l = 2$.

method and the BoneRoute algorithm from Zachariadis et al. [72]. The BoneRoute algorithm is selected for comparison because its proven efficiency and its logic is similar to CBR. Since both algorithms use historical results, it is expected that both resulting solutions should be similar to historical results up to a certain degree.

For the comparison, we solve a VRP with capacity constraint and the objective is modified to consider only total distance minimization. This is because the BoneRoute algorithm was designed for this objective and our CBR method can be easily adapted.

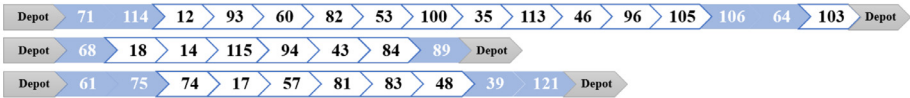
New Instances of 100 customers selected among a total of 125 are solved. The customers in these instances are subsets of the set used in subsection 5.1.

To allow for a fair comparison, a few adaptations were made to both methods. First, both the pool and the case base are generated from the same set of initial solutions. The set is generated using the same algorithms and parameters as those used to create the

initial pool in Zachariadis et al. [72]. Initial solutions are obtained by running the Paessens algorithm [50] and they are further improved using a Tabu Search.

In Table 10, a comparison is made between the results obtained by BoneRoute and the CBR methodology. The columns showing the gaps are calculated using the following logic: $(Value_BoneRoute - Value_CBR)/Value_BoneRoute$. The methods are compared based on their ability to find solutions which reduce the total distance traveled, keep as many customers in chains and keep long chains. As expected, the distances obtained with CBR are 18.4% higher in average. However, the number of customers found in chains is much higher and the length of the chains are also considerably longer. Therefore, the cost for having similar solutions is quite high and the user should be aware of it. However, it is important to recall that some costs are difficult to model and the similarity criterion help to obtain routes that take into consideration these indirect costs.

Routes retrieved:



Final solution:

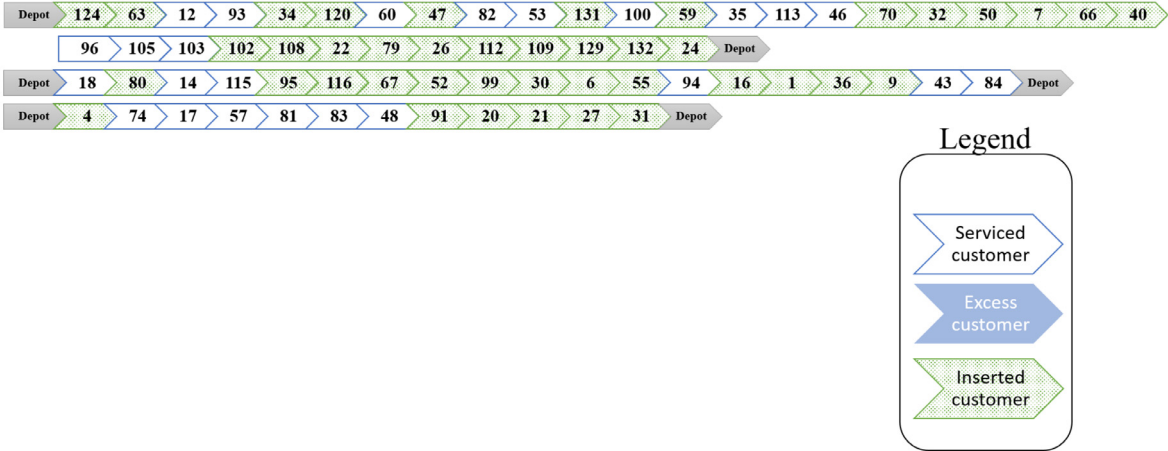


Fig. 10. Comparison of the retrieved routes and the final solution for the real instance *R/3* when using $l = 6$.

Table 10
Comparison of the results obtained with the CBR method and BoneRoute ($n = 40000$, $l = 6$).

ID	Total distance			Number of customers in chains			Longest chain		
	BoneRoute	CBR	% Gap	BoneRoute	CBR	% Gap	BoneRoute	CBR	% Gap
1	1578.7	1948.5	-23.4%	17	92	-441.2%	3	6	-100%
2	1561.4	1843.3	-18.1%	42	95	-126.2%	5	17	-240%
3	1683.7	1905.9	-13.2%	20	89	-345.0%	4	11	-175%
4	1584.7	1929.8	-21.8%	20	71	-255.0%	4	6	-50%
5	1571.5	1890.4	-20.3%	17	82	-382.4%	3	6	-100%
6	1596.7	1923.7	-20.5%	19	87	-357.9%	4	6	-50%
7	1692.1	1914.9	-13.2%	10	67	-570.0%	4	6	-50%
8	1515.8	1683.4	-11.1%	30	91	-203.3%	4	18	-350%
9	1741.3	2184.6	-25.5%	19	85	-347.4%	3	17	-467%
10	1666.6	1949.7	-17.0%	14	68	-385.7%	3	6	-100%

5.5. Real life based case study

In this section, we test the methodology by using real data from an on-line retailer in China. The data set obtained comprises a list of transactions with time and location. Since the size and the volume of the goods delivered as well as the time windows are not available, these constraints will be relaxed for this part of the experiments. The routes come from multiple vehicles covering the same geographical zone in a city. The civic addresses were clustered into small areas including one to four street blocks. These areas act as the customers for the methodology developed. Distances between the customers are calculated according to the definition of the Manhattan distances.

After cleaning the data, a total of 366 routes covering 134 customers are available to build the case base. The number of customers per route range from 3 to 20 customers with the average being 8.07 customers. The size of the case base is considerably less than what was used for the training. It is thus expected that the value of l providing the best results should be a little lower.

To generate test instances, we have randomly selected five days from the data set. For each instance, the routes corresponding to the day selected were removed from the case base. Then, cus-

Table 11
Values of θ_1 obtained with real instances.

ID	Size	l				
		2	3	4	5	6
<i>R/1</i>	52	87.97%	65.38%	29.96%	1.02%	1.02%
<i>R/2</i>	58	78.36%	36.32%	28.25%	3.75%	3.75%
<i>R/3</i>	66	95.65%	61.09%	19.13%	11.69%	5.23%
<i>R/4</i>	78	86.26%	78.60%	46.88%	28.39%	6.08%
<i>R/5</i>	39	87.21%	52.82%	24.56%	5.25%	1.54%

tomers serviced in these days are used to generate the new problems.

Values of θ_1 and θ_2 obtained from the methodology are respectively presented in Table 11 and 12. For θ_1 , the best values obtained range between 86.26% and 95.65%. For θ_2 the best values obtained range from 46.91% to 58.89%. Intermediate results are obtained with $l = 3$ where average values of θ_1 is 58.84% and θ_2 , 38.59%. Concerning the computational run time, it was less than a few seconds due to the limited size of the case base.

The results show that even with a small n , it is still possible to obtain routes similar to new problems. The methodology should

Table 12
Values of θ_2 obtained with real instances.

ID	Size	l				
		2	3	4	5	6
R11	52	25.77%	41.87%	46.91%	46.67%	46.67%
R12	58	24.39%	36.43%	40.39%	56.58%	56.58%
R13	66	25.54%	38.97%	47.17%	54.46%	58.89%
R14	78	33.92%	37.89%	47.67%	50.24%	48.61%
R15	39	28.50%	37.81%	50.67%	57.94%	44.44%

also be adapted by selecting a lower value of l . In a real-life context, logistic managers can also set minimum threshold for θ_1 and θ_2 below which another algorithm would be used to generate the routes such as BoneRoute or any other algorithm, even though the solution might not be similar to historical results. This would ensure that the method is only used in cases where there is enough experience from the drivers.

In order to explain the impact of l on the final solution, we detail a specific instance based on R13. In Fig. 9, $l = 2$ and 20 routes are retrieved. In Fig. 10, $l = 6$ and 3 routes are retrieved. This instance confirms that a higher value l restricts the number of routes retrieved. At the same time, less customers from the problem can be selected by the MIP formulation, thus many more customers need to be inserted. With less routes retrieved, the number of customers in each of these routes is also higher. In this specific instance, it is important to recall that no capacity constraints are enforced which explain why so many customers could be inserted without creating new routes.

6. Conclusion

In this paper, we addressed a problem faced by an on-line retailing company in China in their last-mile delivery. We developed a methodology based on CBR to take into account drivers' experience. A retrieval mechanism based on a MIP formulation allows to retrieve routes that are similar to the new problem from a case base. Then, then retrieved routes are used to produce a solution to a new problem.

The performance of the methodology is first validated using instances generated by a construction heuristic. Average values of $\theta_1 = 67.29\%$ and $\theta_2 = 54.30\%$ show that similar cases can be retrieved. It was also found that these results can be obtained with a maximum increase cost increase of 15%. Thus the CBR methodology succeeds in generating new routes inspired by drivers' experience. This conclusion is also validated by a comparison of results obtained by the BoneRoute algorithm from the literature and the CBR method. Even though a 18.4% cost increase was observed, the results obtained with CBR are considerably more similar to the routes retrieved from the database. The cost increase should also be taken with caution since the results obtained with BoneRoute are so different from what drivers are used to that it is likely that more time will be required to perform the new routes. The performance of the methodology is further validated using real life data obtained from the on-line retailer. In this case, the average value of θ_1 is 58.84%, and the value of θ_2 is 39.38%. Since we have obtained a very limited number of actual routes, the values θ_1 and θ_2 are relatively low. For practical use for the retailer, the performance of the methodology could be easily improved by inserting more routes into the case base. Since the routes in the case base were performed in the past, we can affirm that the new routes are designed based on drivers' experience.

The managerial insight of the paper is twofold: 1) past experience can be formalized, stored in the case base and passed to solve the new problem; 2) the decision-making process for the delivery

routes can be standardized and kept uniform for drivers with different levels of experience.

Traffic congestion is one of factor among others that influence drivers' decisions. Knowing that traffic congestion is highly related to time, in our future work we intend to develop a methodology that will consider specifically the time in relation to the experience of the drivers.

CRedit authorship contribution statement

Olivier Quirion-Blais: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft, Visualization. **Lu Chen:** Conceptualization, Writing - review & editing, Supervision, Funding acquisition.

Acknowledgment

The authors would like to thank the editor and the referees for their valuable comments and suggestions.

References

- [1] Aamodt A, Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun* 1994;7(1):39–59.
- [2] Baldacci R, Mingozzi A, Roberti R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *Eur J Oper Res* 2012;218(1):1–6. doi:10.1016/j.ejor.2011.07.037.
- [3] Bates J, Polak J, Jones P, Cook A. The valuation of reliability for personal travel. *Transportation Research Part E: Logistics and Transportation Review* 2001;37(2):191–229. doi:10.1016/S1366-5545(00)00011-9.
- [4] Beddoe G, Petrovic S, Li J. A hybrid metaheuristic case-based reasoning system for nurse rostering. *Journal of Scheduling* 2009;12(2):99–119. doi:10.1007/s10951-008-0082-8.
- [5] Bouhana A, Fekih A, Abed M, Chabchoub H. An integrated case-based reasoning approach for personalized itinerary search in multimodal transportation systems. *Transportation Research Part C: Emerging Technologies* 2013;31:30–50. doi:10.1016/j.trc.2013.02.014.
- [6] Braekers K, Ramaekers K, Nieuwenhuysen IV. The vehicle routing problem: state of the art classification and review. *Computers & Industrial Engineering* 2016;99:300–13. doi:10.1016/j.cie.2015.12.007.
- [7] Bridge D, Goker MH, McGinty L, Smyth B. Case-based recommender systems. *Knowledge Engineering Review*, 20(3) 2005:315–20. doi:10.1017/S0269888906000567.
- [8] Bräysy O, Gendreau M. Vehicle routing problem with time windows, part I: route construction and local search algorithms. *Transportation Science* 2005;39:104–18. doi:10.1287/trsc.1030.0056.
- [9] Bräysy O, Gendreau M. Vehicle routing problem with time windows, part II: metaheuristics. *Transportation Science* 2005;39(1):119–39. doi:10.1287/trsc.1030.0057.
- [10] Campelo P, Neves-Moreira F, Amorim P, Almada-Lobo B. Consistent vehicle routing problem with service level agreements: a case study in the pharmaceutical distribution sector. *Eur J Oper Res* 2019;273(1):131–45. doi:10.1016/j.ejor.2018.07.030.
- [11] Chakraborty S, Borai S. A developed case-based reasoning system for machine tool selection. *Benchmarking: An International Journal* 2017;24(5):1364–85. doi:10.1108/BJI-07-2016-0103.
- [12] Cordeau J-F, Mäischberger M. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research* 2012;39(9):2033–50. doi:10.1016/j.cor.2011.09.021.
- [13] Cunningham P. CBR: Strengths and weaknesses. In: *11th international conference on industrial and engineering applications of artificial intelligence and expert systems IEA-98-AIE, II*; 1998. p. 517–24.
- [14] Dantzig GB, Ramser JH. The truck dispatching problem. *Manage Sci* 1959;6(1):80–91. doi:10.1287/mnsc.6.1.80.
- [15] Desaulniers G, Madsen OB, Ropke S. Chapter 5: the vehicle routing problem with time windows. In: Toth P, Vigo D, editors. *Vehicle Routing. Society for industrial and applied mathematics (SIAM)*; 2014. p. 119–59. doi:10.1137/1.9781611973594.ch5.
- [16] Dia H. An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transportation Research Part C: Emerging Technologies* 2002;10(5):331–49. doi:10.1016/S0968-090X(02)00025-6.
- [17] Eksioğlu B, Vural AV, Reisman A. The vehicle routing problem: a taxonomic review. *Computers & Industrial Engineering* 2009;57(4):1472–83. doi:10.1016/j.cie.2009.05.009.
- [18] Feillet D, Garaix T, Lehuédé F, Péton O, Quadri D. A new consistent vehicle routing problem for the transportation of people with disabilities. *Networks* 2014;63(3):211–24. doi:10.1002/net.21538.
- [19] Feng T, Arentze T, Timmermans H. Capturing preference heterogeneity of truck drivers' route choice behavior with context effects using a latent class model. *European Journal of Transport and Infrastructure Research* 2013;13(4):259–73.

