CIRRELT

CIRRELT-2019-45

Managing in Real-Time a Vehicle Routing Plan with Time-Dependent Travel Times on a Road Network

Maha Gmira
Michel Gendreau
Andrea Lodi
Jean-Yves Potvin

October 2019

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél. : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél. : 1-418-656-2073
Télécopie : 1-418-656-2624

Managing in Real-Time a Vehicle Routing Plan with Time-Dependent Travel Times on a Road Network

Maha Gmira[1,2,3], Michel Gendreau[1,2], Andrea Lodi[1,2,3], Jean-Yves Potvin[1,4,*]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

[3] Canada Excellence Research Chair in Data Science for Real-Time Decision-Making, Polytechnique Montréal, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

[4] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

Abstract. Devices like geographic information systems, global positioning systems, traffic flow sensors and cellular phones are a source of real-time trac data, for example periodic estimates of travel times in a road network. However, many vehicle routing algorithms do not account for real-time changes in the road network. Accordingly, this paper considers the problem of adjusting in real-time a time-dependent delivery plan to respond to changes in travel speeds and travel times. This is achieved by either modifying the path used to go from one customer to the next in the road network or by modifying the sequence of customers in the planned routes. We also consider a variant of the problem in which deliveries can be canceled. Computational results obtained by running simulations on a real road network for instances with up to 500 customers are reported and analyzed.

Keywords: Vehicle routing, road network, dynamic travel times, metaheuristics, tabu search.

Acknowledgements. Financial support was provided by the Natural Sciences and Engineering Council of Canada (NSERC), while computing facilities were provided by Compute Canada. This support is gratefully acknowledged.

---

* Corresponding author: Jean-Yves.Potvin@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2019

1 Introduction

Vehicle routing has been a core logistic problem since it was introduced by Dantzig and Ramser [5]. Over the last 60 years, numerous papers have dealt with a large number of variants of this basic problem (see, for example, [28]). While early vehicle routing papers considered problems in which all inputs are static, it is now commonly accepted that, especially in urban areas, travel times do vary during a typical planning horizon, e.g., a day. This has led to a significant body of literature on time-dependent routing problems [8]. Recently, some authors have pointed out the fact that to reflect reality, time-dependent problems should not be defined with respect to the graph of customers, but rather with respect to the underlying road network [2, 11].

It is well known that unexpected events, such as accidents and non-recurrent traffic congestion may have an important impact on travel times observed on a road network. Because of that, delivery plans (i.e., solutions of vehicle routing problems) may not be directly implementable in actual conditions. It therefore becomes necessary to adjust delivery plans to travel times that are observed or forecast using the latest information. The purpose of this paper is to develop an optimization methodology to adjust time-dependent delivery plans in an effective fashion. Such a problem falls into the realm of Dynamic Vehicle Routing Problems (DVRPs), i.e., problems in which inputs are received and updated concurrently with the execution of the planned routes [23].

This paper is organized as follows. In Section 2, we review the main classes of DVRPs. Section 3 then provides a detailed description of the problem at hand. Section 4 presents the solution approach proposed to tackle this problem. In Section 5, we describe the comprehensive computational study that was performed to validate our solution approach and report computational results based on a real road network in Montreal. Finally, we conclude and state future research directions in Section 6.

2 Literature review

As mentioned earlier, the problem that we consider falls under the general category of dynamic vehicle routing problems. For more than twenty years, several researchers have been interested in this class of problems, thanks in particular to (1) the increase in computing power due to better hardware, including parallel hardware, and (2) the advent of metaheuristics that allow high-quality solutions to be obtained in reasonable computation times.

Several surveys and book chapters have dealt with DVRPs. A detailed taxonomy of these problems has been presented in [24] based on different criteria: (1) problem type, (2) logistics context, (3) transportation mode, (4) objective function, (5)

fleet size, (6) time constraints, (7) vehicle capacity constraints, (8) ability to reject customers, (9) source of dynamism, (10) source of stochasticity, and (11) solution method. In [1], the authors focus on DVRPs with deterministic and/or stochastic data, where dynamic data refer to new customer locations, customer demands and travel times. In [21], DVRPs are classified according to information quality and evolution. The authors also review various notions relative to the degree of dynamism. The latter is based on 1) the ratio between the number of dynamic requests and the total number of requests, 2) the normalized average of disclosure times, and 3) the reaction time, which is the difference between the disclosure time of a request and the end of its time window. Other interesting survey papers can be found in [14, 15].

In the following, we focus on DVRPs where the source of dynamism either comes from new customer requests or from perturbations to the travel times. It should be noted that, according to [24], about 80% of the papers published on DVRPs involve dynamic customer requests, while about 10% involve dynamic travel times. Only a few papers address other sources of dynamism, like vehicle breakdowns, cargo damage, etc.

2.1 Dynamic customer requests

Several papers have been published in the literature about vehicle routing problems where customer requests occur dynamically over the day and must be handled in real-time. Therefore, rather than an exhaustive review, we provide here a brief overview of how these studies have evolved over time.

At the beginning, purely reactive approaches were proposed to address these problems. Essentially, a new static problem is defined and solved each time a new request occurs (this is called periodic reoptimization in [21]). A good example is the work in [9] in which the authors tackle a courier service application where small parcels are collected at different customer locations and brought back to a central office. The authors have adapted a tabu search with adaptive memory, previously developed for a static version of the problem, to account for the occurrence of new dynamic requests. To allow quick response times, a parallel implementation is proposed. In a subsequent work [13], the diversion of a vehicle from its current destination is considered whenever a new request occurs in the vicinity of the vehicle's current location.

One way to alleviate the myopic behavior of reactive approaches is to accumulate a certain number of dynamic requests before proceeding to reoptimization. This is the approach reported in [19]. In this work, the horizon is divided into time intervals of equal duration and all requests that occur during a given time interval are handled only at the end of that interval using an ant colony algorithm. Another similar approach is found in [25], where new requests are not handled immediately but only after some delay.

A new trend has then emerged, aimed at exploiting any available stochastic information about future customer requests. This information is either considered implicitly or explicitly. In the former case, we refer to the double horizon approach reported in [18], where a different objective is used to optimize the last part of a planned route. Through this objective, time slacks are introduced into the routes to accommodate potential future requests. In the latter case, we refer to the multiple plan approach [3] where different solutions are considered depending on different scenarios for the occurrence of new requests. Another approach to account for future requests is to devise waiting strategies for the vehicles to catch any potential request that can occur in their vicinity, see for example [18].

2.2 Dynamic travel times

The literature related to vehicle routing with dynamic travel times is relatively scarce, when compared to dynamic customer requests. In [7], the authors tackle a dynamic single load pickup and delivery problem. They present a routing system to dispatch a fleet of vehicles according to customer requests that occur at random over a planning horizon. The system exploits online information about travel times from a traffic management center, while also considering the occurrence of new customer requests. In this problem, the travel times are time-dependent and are modeled using a combination of piecewise and smoothing functions to avoid discontinuities [6]. Due to the computational load, only a subset of possible paths between customers are monitored when changes to travel times occur.

In [27], a genetic algorithm is proposed to address the same type of problem. A traffic simulator is used to model changes in travel times due to traffic incidents. The algorithm was applied to a small test network with 25 nodes and 80 arcs. The results indicate that the total travel time is improved on average by 3.7% over an approach where the travel times are forecast in advance.

In [12], the authors address a dynamic pick-up or delivery vehicle routing problem with time windows and time-dependent travel times. They consider multiple vehicles with different capacities, dynamic service requests, and dynamic variations in travel times. First, the problem is formulated as a mixed integer linear programming model and solved exactly for small instances with up to 10 customers. Larger instances are solved with a genetic algorithm. The results show that accounting for dynamic travel times is worthwhile, even when all requests are known in advance, in scenarios where the travel times fluctuate significantly over the planning horizon.

In [22], a tolerance for lateness at each customer is introduced. When the delay exceeds the tolerance, reactive changes are applied to the existing routes. This work was then extended in [17] by allowing diversion of vehicles from their planned destination, thus opening up additional opportunities to serve new requests. In [26], a

further extension is considered, where it is assumed that the current position of each vehicle is known at all time, not only when a new request occurs or when a vehicle arrives at a customer location.

In [29], travel times are derived through a queuing model for traffic flows, where vehicles are "served" by road segments. The queuing model captures the relationship between traffic flow, density and speed, from which travel times can be obtained. Dynamic perturbations to the travel times are then produced by changing the inputs of the queuing model. This queuing approach is compared with alternative approaches and its benefits are evaluated.

In [4], a technician routing problem with stochastic service and travel times is introduced and solved using a two-stage stochastic programming method. The authors consider a specific variant of the field service routing problem in which two types of customers are served: mandatory, i.e., they have to be served within a specified time window, and optional, i.e., they may be served (or not) within the planning horizon.

3 Problem Description

In this section, we first describe the planning problem that is used to generate the planned routes at the beginning of the day, before introducing its dynamic version in which the time-dependent delivery plan is adjusted in real-time to respond to changes in travel times. A variant of this problem is also considered in which some deliveries can be canceled.

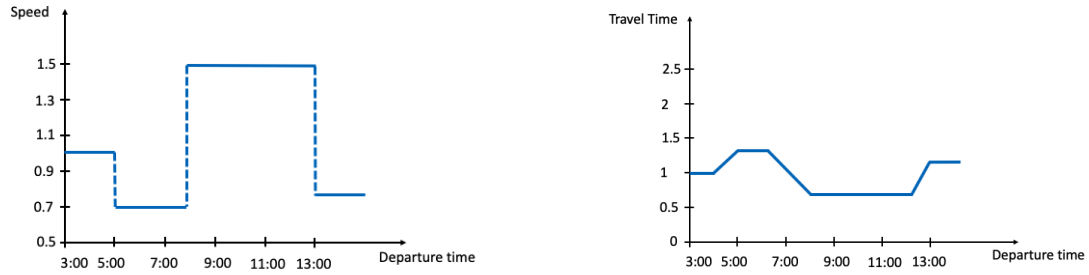
3.1 Planning problem

We consider a vehicle routing problem with time windows and time-dependent travel times defined on a road network. In the following, we describe the road network and the time-dependent travel times, before formalizing the problem.

Road Network. The road network is a directed graph $G = (V, E)$, where the set of vertices $V = \{0, 1, 2, ..., n\}$ corresponds to road junctions and the set of arcs E to road segments between pairs of junctions. Each arc (i, j) is associated with a distance d_{ij}, a time-dependent speed function $v_{ij} : t \rightarrow R^+$ that returns the speed at time t, and a time-dependent cost function $c_{ij} : t \rightarrow R^+$ that returns the cost of traversing arc (i, j) at time t (often, c_{ij} corresponds to the travel time).

Travel Times. With each arc is associated a stepwise time-dependent travel speed function, where a different speed is associated with each time interval defined over

the planning horizon. From this time-dependent travel speed function, a piecewise linear time-dependent travel time function can be derived, see Fig. 1.



(a) Speed function

(b) Travel time function

Figure 1: Piecewise linear travel time function derived from the stepwise speed function for an arc of length 1

A time-dependent variant of Dijkstra's algorithm can then use these arc-based functions to compute shortest paths (with regard to travel time) between any pair of nodes for any departure time [11].

Problem. Our problem involves a set of customers $C \subset V$ and a depot (node 0) located on the road network. Each customer $i \in C$ has a demand q_i, a time window $[a_i, b_i]$ for the service start time, where a_i and b_i are the lower and upper bounds of the time window, respectively, and a service or dwell time s_i. A set of vehicles K, each of capacity Q, is located at the depot. A vehicle cannot arrive at customer i after the upper bound b_i of the time window, but can arrive before the lower bound a_i, in which case the vehicle waits until time a_i to start the service. The time window at the depot $[a_0, b_0]$ defines the beginning and end of the planning horizon. The goal is to generate a set of feasible vehicle routes (solution), one for each vehicle, that start and end at the depot and serve all customers at minimum cost, where the cost corresponds to the total duration of the routes (i.e., travel time + waiting time + dwell time). A solution is feasible if it satisfies the capacity constraints and time windows. A solution is obtained by solving the planning problem with the tabu search heuristic reported in [11], where the upper bound of the time windows at each customer and the end of the time horizon cannot be exceeded (i.e., no lateness and no overtime are allowed). The number of vehicles is a decision variable in the planning problem.

3.2 Dynamic problem

The dynamic version of the problem is obtained by allowing dynamic perturbations to the time-dependent travel speeds. When a perturbation impacts one or more arc

at a given time, the time-dependent travel speed function of each arc is modified, as indicated by the dotted red lines in Fig. 2.

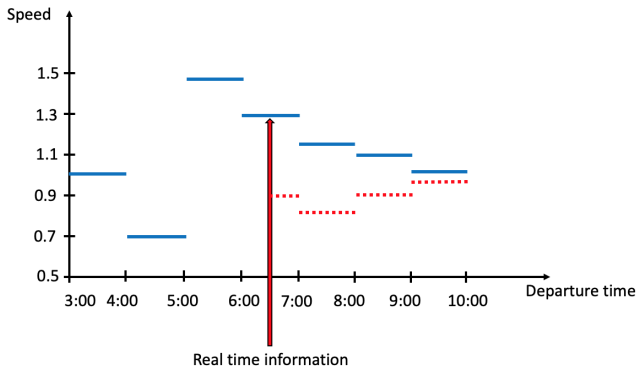


Figure 2: Example of a dynamic perturbation to a travel speed function

Clearly, if one of these arcs is part of the path leading from one customer to the next in a delivery route, the travel time between these two customers will change, with further impacts for the customers down the route, since the travel times are time-dependent. This is illustrated in Fig. 3, where a vehicle route starts from the depot, and while it is en route to its current destination (customer C_4), a travel speed update occurs, as indicated by the black dot. The system reacts by first recalculating the new arrival and departure times at customer C_4 from the current vehicle's position. This is repeated for each customer along the route, including the end depot. Eventually, it may be that the best path to reach the next location, starting with the current vehicle's position, will need to be recalculated, as it is illustrated in the figure. It may even be that the sequence of customers along the route has to be modified to account for the perturbation, as it is explained in Section 4.

In the dynamic version, it is not possible to guarantee anymore that the service will take place within the customers' time windows or that a vehicle will return to the depot before the end of the time horizon. Accordingly, the objective function of the dynamic problem minimizes the sum of travel time and a lateness penalty. The penalty is linear and is obtained by multiplying the total lateness by a penalty factor.

A variant of the dynamic problem is also considered where it is possible to cancel a customer if the lateness at this customer exceeds a given threshold. In this case, a large fixed penalty for each cancellation is added to the objective.

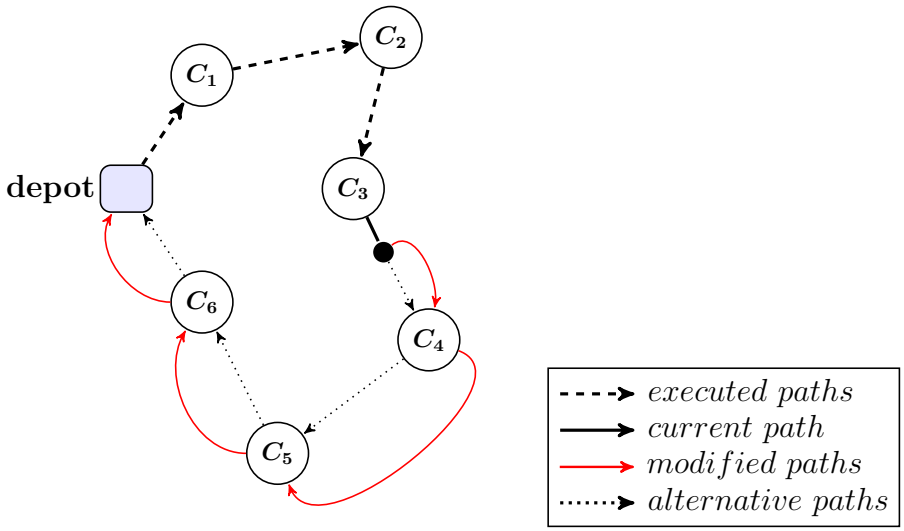


Figure 3: Dispatching of one vehicle in a dynamic setting (at the time of a new travel time update)

4 Dispatching system

The operations of the real-time dispatching system are based on the following assumptions:

- Real-time changes involve only travel speeds;
- Travel speed updates can occur at any time and involve a subset of arcs in the road network;
- The dispatching system can communicate at any time with a vehicle;
- The dispatching system knows the current location of each vehicle;
- When a customer is visited, this customer is removed from the planned route;
- Each time a travel speed update occurs, the current planned routes are reconsidered (which may imply reoptimization);
- During reoptimization, the exchange of customers between two routes is not allowed, since it is a delivery problem;
- During reoptimization, the current destination (customer) of each vehicle cannot change. Thus, no diversion is allowed.

The main components of the dispatching system are illustrated Fig. 4.

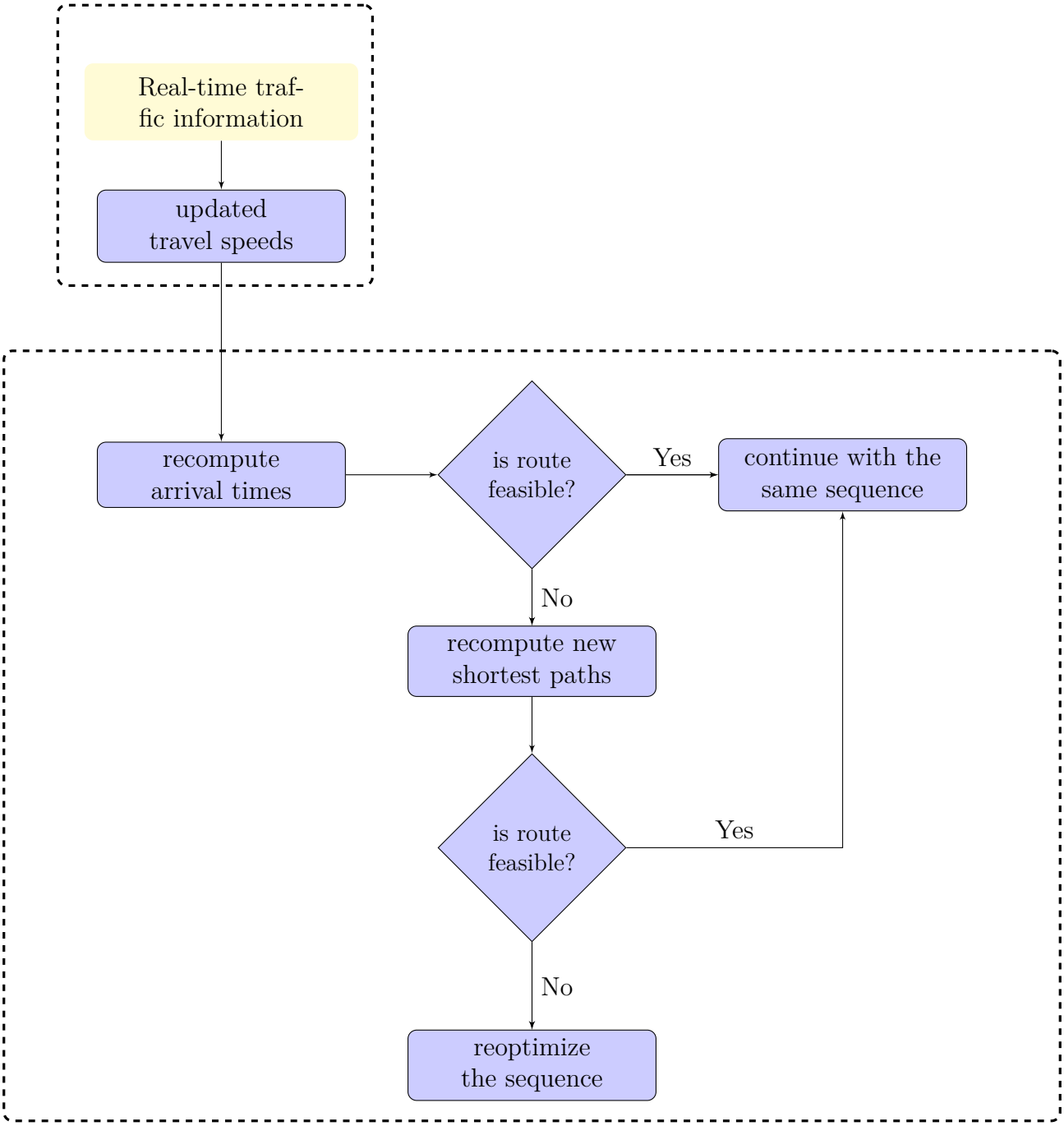


Figure 4: Real-time dispatching system

We assume that we are working in an environment in which a first delivery plan was obtained by solving the planning problem described in section 3.1. As time goes on, new information on current and forecast travel times becomes available and needs to be acted upon.

While the number of vehicles is a decision variable in the planning problem, it becomes fixed and cannot change in the dynamic setting. From the solution of the planning problem, we can compute the latest departure time at each customer in each route, which is a bound that guarantees feasibility of an entire planned route if the departure time from a customer in that route does not exceed its corresponding bound. These bounds are updated each time the solution is modified.

Once a speed update has been received, the planning problem that corresponds to the current status of the planned routes is addressed. Each vehicle route is considered in turn and new arrival and departure times are calculated at each customer. If the route is still feasible, then we keep it as it is. Otherwise, we recompute the best path between two consecutive locations in the route, starting with the vehicle's current location and its current destination and ending with the last customer and the end depot. As previously mentioned, this is done with a time-dependent variant of Dijkstra's algorithm.

Now, if the route is still infeasible, we try to improve it by modifying the sequence of customers (with the exception of the vehicle's current destination which is fixed) by applying a local search descent based on the Or-opt neighborhood [20]. That is, sequences of three, two and one consecutive customers are removed from the route and reinserted at the best possible place. This neighborhood is explored in a systematic way by first considering all possible sequences of three consecutive customers, two consecutive customers and a single customer, in this order. An example of an Or-opt with three customers is illustrated in Figure 5.

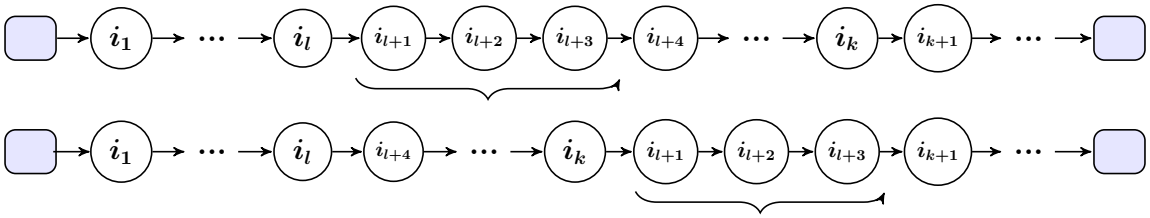


Figure 5: An Or-opt exchange of three customers

In this example, the sequence of customers i_{l+1}, i_{l+2}, i_{l+3} is removed from the route and reinserted at another place, here between customers i_k and i_{k+1}. One advantage of Or-opt exchanges is that they do not reverse any part of the route, which is a desirable characteristic for problems with time windows. Clearly, once the sequence of customers is modified, the best path between two consecutive customers needs to be

recalculated, starting with customer i_l in the figure. The final route obtained, which is a local minimum with regard to the Or-opt neighborhood, is then kept (feasible or not).

To summarize, we can distinguish three different cases when a reaction to a speed update occurs:

Case 1: the planned route remains feasible and the shortest paths and sequence of customers remain the same.

Case 2: the planned route is infeasible and feasibility is restored with changes to the shortest paths.

Case 3: the planned route is infeasible and feasibility is not restored with changes to the shortest paths.

Case 3.1: reoptimization with Or-opt restores feasibility of the planned route.

Case 3.2: reoptimization with Or-opt does not restore feasibility of the planned route.

5 Computational Study

In the following, we describe the comprehensive computational study that is used to validate our solution approach. We first describe the simulator developed to generate travel speed updates. Then, we introduce the test instances, which are based on a real road network. Finally, we report results obtained with our methodology with an increasing number of customers. The code is written in Java and the experiments were performed on a Dell PowerEdge R630 server with two Intel Xeon processors E5-2637V4 with 4 cores and 128GB of memory each.

5.1 Simulator

We have developed a discrete-event simulator to generate travel speed updates and to maintain the status of the current solution (current location of each vehicle, current planned routes). A reaction takes place when speed updates occur. Their occurrence is modeled through an exponential distribution with an average inter-arrival time of 10 minutes.

The time-dependent travel speed function associated with each arc in the road network is a stepwise function, where a different speed is defined for each one-hour time interval over a planning horizon that extends from 9:00 AM to 7:00 PM. These

travel speed functions are used to produce the initial routes at the beginning of the day and will be referred to as basic speed functions.

To generate travel time perturbations, we first randomly select one of the 12 districts that constitute the three central boroughs of Montreal used for this study (see Section 5.2). Then, the travel speed function of each arc (road segment) in the selected district is modified by applying a congestion rate between 1 and 5 to the basic speed in the corresponding time interval. The new speed is obtained by dividing the basic speed by the congestion rate. Thus, a lower rate stands for a minor incident while a higher rate stands for a major one.

It is assumed that an incident that occurs during a peak hour will have a major impact. Accordingly, the selection of the congestion rate of each arc is done probabilistically, using a probability distribution defined over the intervals [1, 2[, [2, 3[, [3, 4[and [4, 5], with a bias towards interval [4, 5]. Conversely, if an incident occurs during an off-peak hour, the probability distribution is biased towards the interval [1, 2[. The time span of an incident is also determined by the congestion rate. For example, if the latter is in the interval [4, 5], the basic speeds of the next three one-hour time intervals are also modified. If the congestion rate is smaller, then the number of affected time intervals is also smaller.

5.2 Test instances

The travel speed data used to generate the basic speed function of each arc in the road network were provided by a software development company that produces vehicle routing algorithms to plan the home delivery of large items (appliances, furnitures) to customers. Based on historical data that span approximately two years of operations, a previously developed neural network model [10] was used to mine this information and produce basic travel speed functions for a typical operations day. These functions were then used to produce the initial planned routes at the beginning of the day.

The computational study is based on three central boroughs of Montreal with an important commercial activity and that are particularly sensitive to congestion, namely, Plateau Mont-Royal, Ville-Marie and Outremont, as shown in Fig. 6. These three boroughs contain over $4,400$ road segments (arcs) and $2,150$ road junctions (nodes). Plateau Mont-Royal has 6 districts, while Ville Marie and Outremont have 3 districts each, for a total of 12 districts.

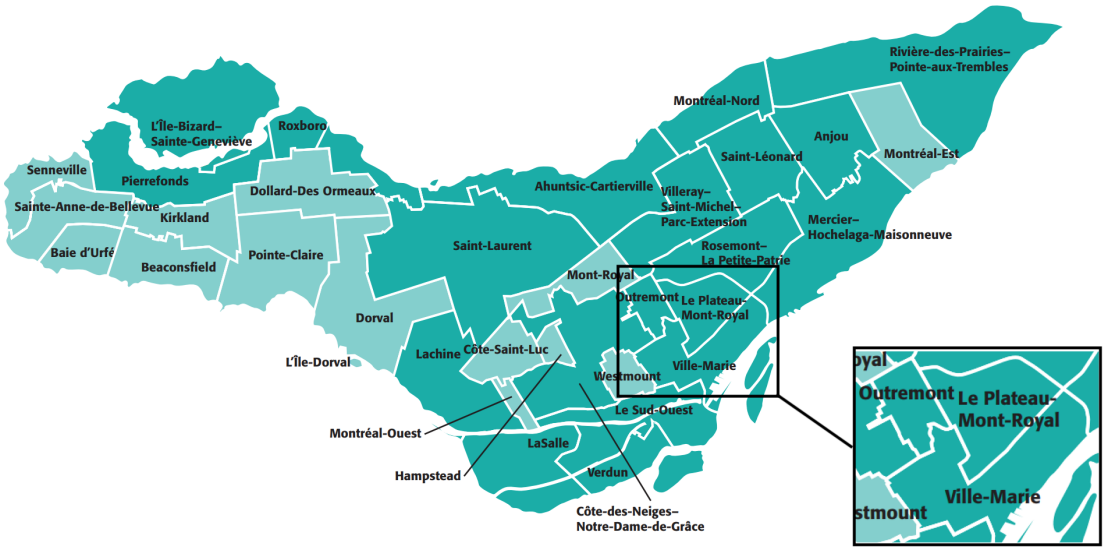


Figure 6: The three central boroughs used for the study

A central depot and a number of customers varying between 100 and 500 were randomly chosen in the above road network. More precisely, we have created five instances each with 100, 200, 300, 400 and 500 customers, for a total of 25 instances. For each customer i, the lower bound of the time window a_i was randomly chosen inside the planning horizon (while allowing sufficient time to return the depot from the customer). Narrow and wide time windows were produced according to [16], by setting the upper bound b_i in minutes to $(a_i + 3 \leq b_i \leq a_i + 4)$ and $(a_i + 10 \leq b_i \leq a_i + 15)$, respectively, while always taking care not to exceed the end of the planning horizon. The capacity of each vehicle was set to the number of customers in the instance multiplied by 8 (which means 800 for instances of size 100 and 4,000 for instances of size 500). The customer demands were generated between a lower bound of 20 and an upper bound of 70, by following the procedure reported in [2]. Finally, and without loss of generality, there is no service or dwell time at each customer.

5.3 Results

In the following, we report computational results obtained with our reactive procedure. With regard to the objective function used in these experiments (total route duration plus total lateness at customers plus total overtime at the end depot plus, possibly, total cancellation penalty), it should be noted that the lateness at each customer and the overtime at the end depot were multiplied by a penalty factor equal

to 5 and that the cancellation of each customer led to a penalty of 1,000.

For comparison purposes, we consider an alternative non-reactive procedure which is the following. We consider the planned routes produced at the beginning of the day and, while travel speed updates occur along the day, we keep following the same planned routes and keep using the same path in the road network to go from one customer to the next. Thus, we do not react to the dynamic speed updates and we evaluate the objective value of this solution at the end of the simulation.

The results are reported in tables 1 and 2 and represent averages over the five instances of each size. The columns are the following:

- #Cust.: number of customers;
- #Cust./Route: number of customers per route;
- Scen.: in scenario 1, only lateness at customers is allowed; in scenario 2, cancellations are also allowed when the lateness exceeds a given threshold (set to 30 minutes);
- Comp. Time(s): total computation time in seconds for reacting to all speed updates in the dynamic setting;
- $Impr_1$: improvement in percentage of the objective value of the solution produced by the reactive procedure over the one produced by the non-reactive procedure, when starting with the initial feasible static solution:

$$Impr_1 = \frac{Non\text{-}Reactive - Reactive}{Non\text{-}Reactive} \times 100, \tag{1}$$

- $Impr_2$: improvement in percentage of the objective value of the solution produced by the reactive procedure over the one produced by the non-reactive procedure, when starting with a static solution that allows lateness and overtime (i.e. the objective used to generate the initial solution is total duration of the routes plus total lateness plus total overtime, where the last two components are multiplied by the penalty factor); this improvement is calculated as in equation (1);
- Total Dur.: total route duration in minutes over all routes;
- Total Lat.: sum of lateness in minutes over all customers (thus, excluding the end depot);
- Total Over.: sum of overtime at the end depot in minutes over all routes;

- # Canc.: number of customer cancellations;
- # Late Cust.: number of customers that are served late.

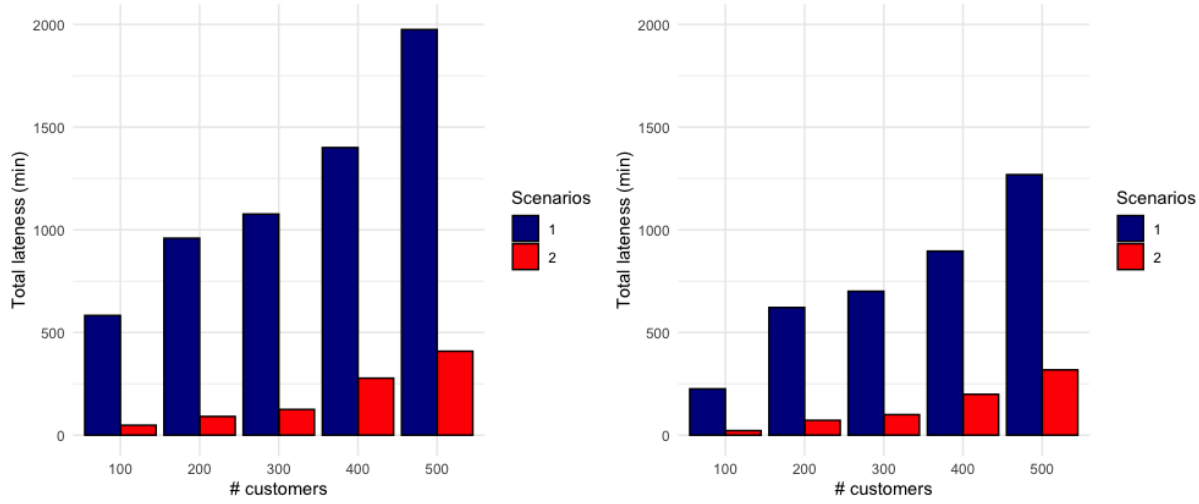
Table 1: Results for instances with narrow time windows

# Cust.	# Cust. /Route	Scen.	Comp. Time (s)	Total Dur.	Total Lat.	Total Over.	$Impr_1$ (%)	$Impr_2$ (%)	# Canc.	# Late Cust.
100	20	1	599.6	630.0	583.0	84.1	2.86	2.40	-	32
		2	421.5	582.2	48.3	0	9.64	-	8	2
200	38	1	817.3	1231.6	959.1	100.4	2.17	1.78	-	61
		2	406.4	1099.6	91.0	23.7	8.83	-	22	9
300	63	1	1427.6	1706.6	1076.6	132.5	2.25	1.91	-	85
		2	740.1	1411.4	125.2	53.4	8.34	-	28	15
400	79	1	2182.4	2227.8	1400.4	292.4	2.09	1.88	-	88
		2	917.7	1912.8	277.6	66.0	7.06	-	30	33
500	104	1	2985.9	2761.2	1975.4	365.1	2.46	2.07	-	102
		2	1695.8	2349.0	408.7	75.1	7.05	-	35	48

Table 2: Results for instances with wide time windows

# Cust.	# Cust. /Route	Scen.	Comp. Time (s)	Total Dur.	Total Lat.	Total Over.	$Impr_1$ (%)	$Impr_2$ (%)	# Canc.	# Late Cust.
100	18	1	691.1	633.4	225.5	60.5	3.94	3.59	-	28
		2	383.8	585.3	22.1	0	11.83	-	8	2
200	36	1	944.8	1233.3	621.7	87.6	3.74	3.37	-	58
		2	591.4	1125.5	72.3	17.0	9.50	-	18	7
300	59	1	1622.3	1525.4	700.7	90.0	3.74	3.37	-	66
		2	1006.2	1420.4	100.0	39.1	8.29	-	21	9
400	74	1	2486.5	2027.0	896.1	209.2	3.10	2.76	-	89
		2	1140.0	1902.7	198.9	52.2	8.10	-	25	22
500	99	1	3222.3	2525.2	1268.6	246.0	2.95	2.66	-	92
		2	2187.1	2380.5	317.9	61.9	7.13	-	29	37

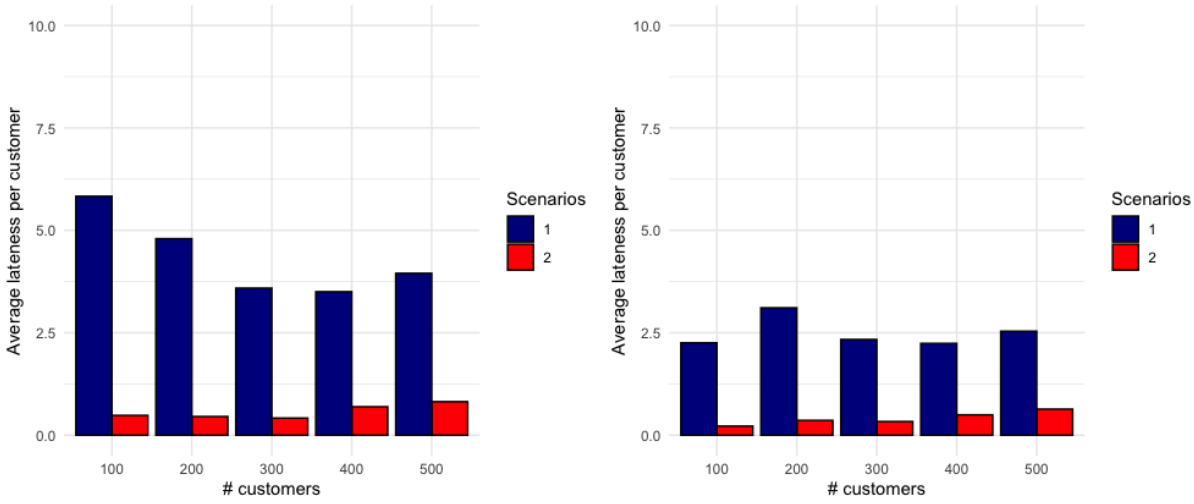
First, it should be noted that the number of customers per route increases with the total number of customers, in such a way that the number of vehicles stays around 5 in all cases. Clearly, the density of customers increases with the number of customers (since the service area remains the same), thus leading to smaller travel times between customers and thus, more customers in the planned routes. Not surprisingly, the total lateness and total overtime increase significantly from the smallest instances with 100 customers to the largest ones with 500 customers, particularly when the time windows are narrow since there is less flexibility for updating the shortest paths and optimizing the sequence of customers in the planned routes. The total lateness and total overtime also decrease significantly when customer cancellations are allowed (reduction factor of 7.5 and 6 for narrow and wide time windows, respectively). These trends are illustrated in Fig. 7 for the total lateness at customers. The average lateness per customer over all customers in the planned routes is also shown in Fig. 8. This average does not vary much with the instance size, but is clearly higher for instances with narrow time windows (overall averages of 4.3 and 2.5 minutes for narrow and wide time windows, respectively, for scenario 1). It also decreases sharply when cancellations are allowed since customers with high lateness can be removed from the planned routes.



(a) Narrow time windows

(b) Wide time windows

Figure 7: Total lateness by number of customers and scenarios



(a) Narrow time windows

(b) Wide time windows

Figure 8: Average lateness per customer by number of customers and scenarios

The improvement $Impr_1$ shows the benefits associated with the reactive procedure over the non-reactive one, when starting with the initial feasible static solution. The overall average improvement for instances with narrow and wide time windows is 5.3% and 6.2%, respectively. The improvement is larger for instances with wide time windows because there is more flexibility for reoptimization. When breaking these averages according to scenarios 1 and 2, we obtain 2.4% and 8.2% for narrow time windows and 3.5% and 9.0% for wide time windows. Thus, the improvement is much larger when cancellations are allowed. This is easily explained since the planned routes produced by the reactive procedure do not contain all customers, due to cancellations, thus leading to shorter routes with smaller total lateness and total overtime. In a sense, the comparison between the two procedures is not fair in this case because all customers are visited with the non-reactive procedure. The improvement $Impr_2$ of the reactive procedure over the non-reactive one, when starting with the initial static solution that allows lateness and overtime, is smaller than $Impr_1$ (by about 15%), since better solutions are to be expected when the time windows are relaxed.

In the case of scenario 2, we can see in tables 1 and 2 that the percentage of canceled customers never exceeds 11% and 9% for narrow and wide time windows, respectively. However, as previously observed, it leads to large improvements with regard to total lateness and total overtime, when compared to scenario 1. Furthermore, the percentage of late customers decreases sharply from 25.3% in scenario 1 to 5.1% in scenario 2.

While Tables 1 and 2 report the total computation time for the reactive procedure

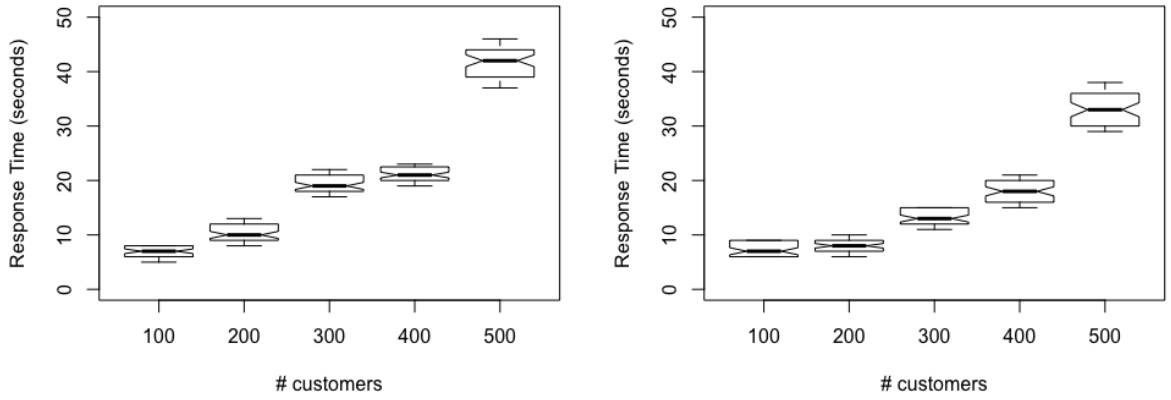
over all speed updates, Tables 3 and 4 now show the average reaction time in seconds per speed update for narrow and wide time windows, respectively. Figure 9 provides an alternative view with boxplots that indicate the minimum, first quartile, median, third quartile and maximum values. As we can see, the maximum reaction time for the largest instances with 500 customers does not exceed one minute. Given that we did not put a full implementation effort to speed up the computations, the reactive procedure appears to be viable in a real-time context.

Table 3: Reaction time - Narrow time windows

# cust.	Min	Max	Mean
100	5	8	6.9
200	8	13	10.1
300	17	22	19.4
400	19	23	21.0
500	37	46	41.6

Table 4: Reaction time - Wide time windows

# cust.	Min	Max	Mean
100	6	9	7.4
200	6	10	8.0
300	11	15	13.2
400	15	21	18.0
500	29	38	32.9



(a) Narrow time windows (b) Wide time windows

Figure 9: Reaction time

5.4 Impact of reoptimization with Or-opt

In this section, we evaluate the benefits of reoptimizing the planned routes with a local descent based on Or-opt exchanges. That is, we compare the original reactive procedure with a variant where the shortest paths can be recomputed but no reoptimization of the sequence of customers with Or-opt takes place. It corresponds to bypassing the rectangle "reoptimize the sequence" in the illustration of the dispatching system in Fig. 4. The results are reported in Tables 5 and 6. In these tables, we show the results of the procedure without reoptimization. We also include the percentage of improvement $Impr_3$ of the procedure with reoptimization over the one without reoptimization, that is:

$$Impr_3 = \frac{Without\ Reoptimization - With\ Reoptimization}{Without\ Reoptimization} \times 100. \qquad (2)$$

Overall, $Impr_3$ is equal on average to 0.74% and 1.01% for narrow and wide time windows, respectively. Although some improvements are obtained with reoptimization, they are relatively modest. We think that these results can be explained by the dynamic setting where frequent speed updates occur, thus quickly making a large part of the reoptimization work obsolete. On the other hand, if we focus on the total lateness and total overtime components of the objective, we observe improvements of 15.5% and 15.6%, respectively, over all test instances. However, this is achieved by increasing the duration of the routes.

Table 5: Results with no reoptimization - Narrow time windows

# Cust.	Scenario	Comp. Time(s)	Total Dur.	Total Lat.	Total Over.	$Impr_3$ (%)	# Canc.	# Late Cust.
100	1	328.6	530.2	676.3	96.7	0.24	-	36
	2	205.3	552.9	62.8	18.5	0.43	8	4
200	1	322.0	1055.0	1130.8	117.5	0.50	-	70
	2	163.0	1084.9	111.9	27.7	0.78	22	11
300	1	456.8	1508.5	1268.2	157.7	0.54	-	98
	2	261.3	1390.7	153.0	62.4	0.84	31	37
400	1	816.2	1932.1	1677.7	345.0	0.75	-	101
	2	368.0	1865.3	338.72	75.9	1.19	32	38
500	1	979.4	2347.3	2372.4	433.4	0.88	-	119
	2	634.2	2222.0	560.0	86.7	1.27	37	54

Table 6: Results with no reoptimization - Wide time windows

# Cust.	Scenario	Comp. Time(s)	Total Dur.	Total Lat.	Total Over.	$Impr_3$ (%)	# Canc.	# Late Cust.
100	1	352.5	592.1	261.6	69.6	0.50	-	32
	2	186.5	575.5	25.7	13.3	1.03	8	3
200	1	453.5	1134.6	721.2	101.3	0.74	-	65
	2	277.9	1122.6	83.8	19.4	0.94	20	8
300	1	746.2	1418.6	812.8	104.8	0.75	-	75
	2	460.9	1416.1	116.0	44.6	1.17	23	11
400	1	1069.2	1872.9	1039.5	244.5	0.80	-	101
	2	501.6	1892.0	230.7	60.0	1.28	27	25
500	1	1256.7	2318.2	1471.6	287.8	0.93	-	105
	2	940.5	2374.9	368.8	71.6	1.97	31	43

5.5 Impact of cancellation threshold

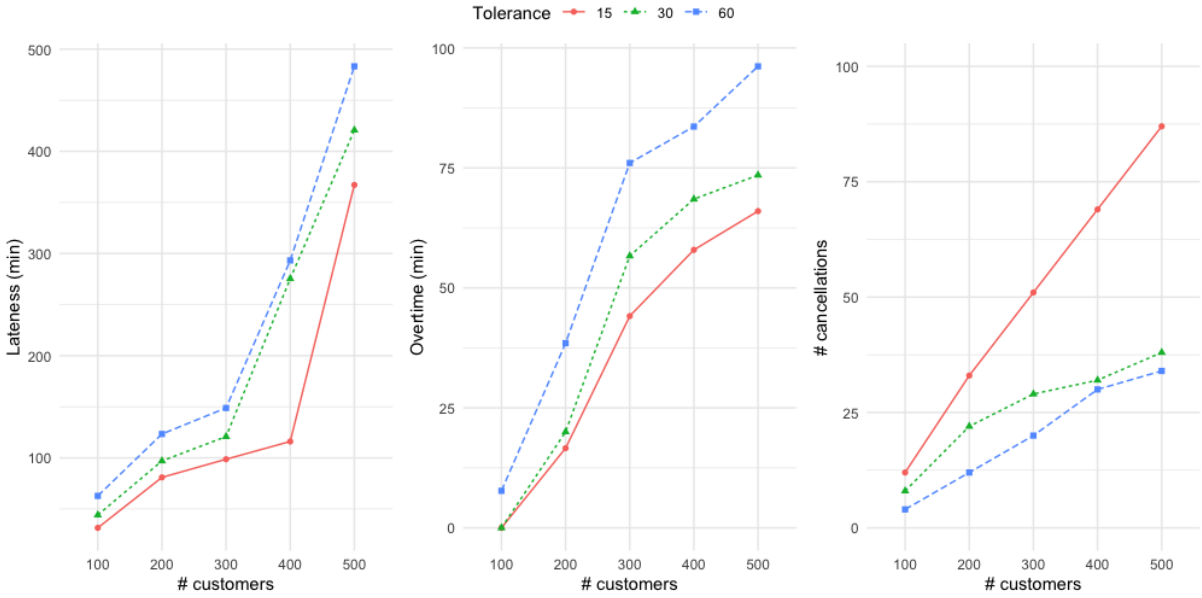
We consider in this section the impact of the cancellation threshold, that is, the lateness value at a customer location beyond which cancellation can be considered, if it is beneficial with regard to the objective function. Tables 7 and 8 show the total lateness, total overtime, and number of cancellations using three different threshold values of 15, 30 and 60 minutes for the instances with narrow and wide time windows, respectively. The trends observed in these tables are also illustrated in Fig. 10. Clearly, the total lateness, total overtime, and number of cancellations increase from the smallest instances with 100 customers to the largest instances with 500 customers. The total lateness and total overtime also increase from the smallest threshold value of 15 minutes to the largest value of 60 minutes, while the number of cancellations decreases. Furthermore, the differences between the solutions obtained with the three threshold values, either with regard to total lateness, total overtime or number of cancellations, tend to increase with the number of customers. The difference in the number of cancellations between a threshold of 15 minutes and a threshold of 30 minutes is particularly noticeable in this regard. Overall, these results indicate that the solutions obtained with our reactive procedure are quite sensitive to the value of this cancellation threshold.

Table 7: Impact of cancellation threshold - Narrow time windows

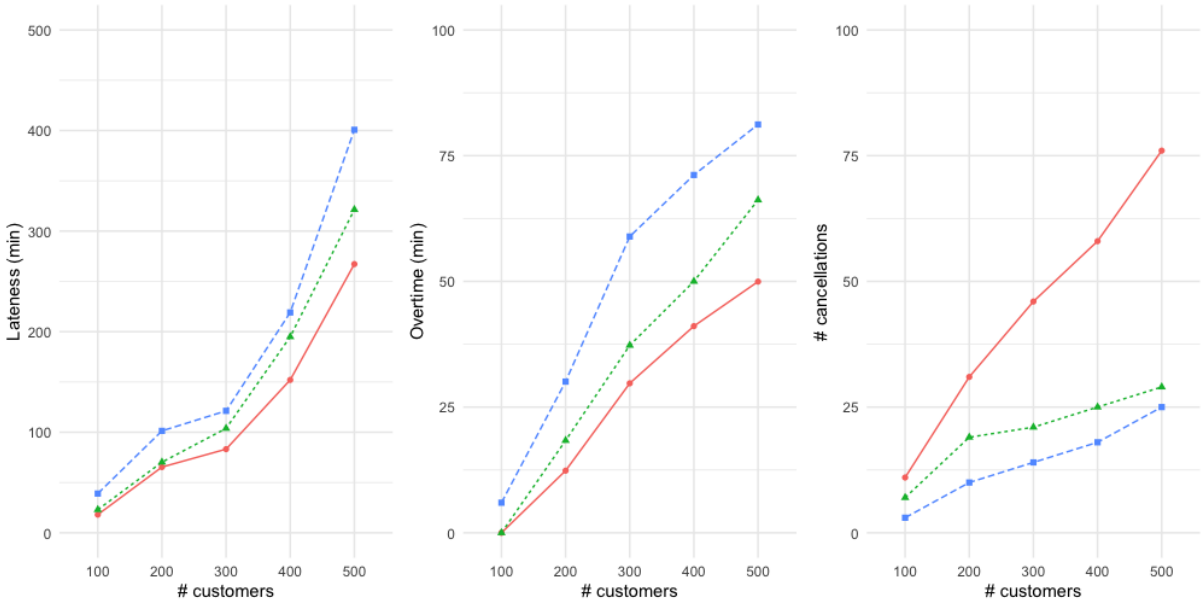
# cust.	Cancellation Threshold	Total Lateness	Total Overtime	# Canc.
100	15	31.50	0	12
	30	44.10	0	8
	60	62.79	7.72	4
200	15	80.91	16.61	33
	30	97.00	20.01	22
	60	123.38	38.49	12
300	15	98.67	44.13	51
	30	120.83	56.66	29
	60	148.84	76.03	20
400	15	116.06	57.93	69
	30	275.34	68.50	32
	60	293.22	83.62	30
500	15	367.11	66.00	87
	30	420.75	73.50	38
	60	483.18	96.17	34

Table 8: Impact of cancellation threshold - Wide time windows

# cust.	Cancellation Threshold	Total Lateness	Total Overtime	# Canc.
100	15	18.10	0	11
	30	23.00	0	7
	60	39.02	5.99	3
200	15	65.38	12.36	31
	30	70.23	18.36	19
	60	101.35	30.07	10
300	15	83.19	29.73	46
	30	103.82	37.28	21
	60	121.26	58.92	14
400	15	152.04	41.10	58
	30	194.92	50.00	25
	60	219.11	71.16	18
500	15	267.21	49.96	76
	30	321.37	66.19	29
	60	400.72	81.20	25



(a) Narrow time windows



(b) Wide time windows

Figure 10: Impact of cancellation threshold

6 Conclusion

In this paper, we address a time-dependent vehicle routing problem with dynamic speed updates. A reactive procedure was devised that accounts for the time-dependent nature of the problem. The results obtained on a real road network show that improvements to the objective value are obtained when reacting to the travel speed updates. With regard to future work, we want to address a similar home delivery problem, but with a heterogeneous fleet. One can imagine larger vehicles that are used in the outskirts of a city, while smaller vehicles that are less sensitive to congestion are used downtown. Hence, different types of vehicles would be associated with different types of time-dependent travel speed functions. This problem can be compounded by allowing larger vehicles to transfer a part of their load to smaller vehicles, thus leading to synchronization issues.

Acknowledgments. Financial support was provided by the Natural Sciences and Engineering Research Council of Canada through its Canada Excellence Research Chair program. Computing facilities were provided by Compute Canada. This support is gratefully acknowledged.

References

[1] T. Bektaş, P.P. Repoussis, and C.D. Tarantilis. Chapter 11: Dynamic Vehicle Routing Problems. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 299–347. SIAM, 2014.

[2] H. Ben Ticha, N. Absi, D. Feillet, A. Quilliot, and T. Van Woensel. A branch-and-price algorithm for the vehicle routing problem with time windows on a road network. *Networks*, 73(4):401–417, 2019.

[3] R.W. Bent and P. Van Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004.

[4] S. Binart, P. Dejax, M. Gendreau, and F. Semet. A 2-stage method for a field service routing problem with stochastic travel and service times. *Computers & Operations Research*, 65:64–75, 2016.

[5] G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959.

[6] B. Fleischmann, M. Gietz, and S. Gnutzmann. Time-varying travel times in vehicle routing. *Transportation Science*, 38(2):160–173, 2004.

[7] B. Fleischmann, S. Gnutzmann, and E. Sandvoß. Dynamic vehicle routing based on online traffic information. *Transportation Science*, 38(4):420–433, 2004.

[8] M. Gendreau, G. Ghiani, and E. Guerriero. Time-dependent routing problems: A review. *Computers & operations research*, 64:189–197, 2015.

[9] M. Gendreau, F. Guertin, J.-Y. Potvin, and É. Taillard. Parallel tabu search for real-time vehicle routing and dispatching. *Transportation Science*, 33(4):381–390, 1999.

[10] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin. Travel speed prediction based on learning methods for home delivery. Technical report, CIRRELT-2018-46, Montréal, Canada, 2018.

[11] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. Potvin. Tabu search for the time-dependent vehicle routing problem with time windows on a road network. Technical report, CIRRELT-2019-32, Montréal, Canada, 2019.

[12] A. Haghani and S. Jung. A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research*, 32(11):2959–2986, 2005.

[13] S. Ichoua, M. Gendreau, and J.-Y. Potvin. Diversion issues in real-time vehicle dispatching. *Transportation Science*, 34(4):426–438, 2000.

[14] S. Ichoua, M. Gendreau, and J.-Y. Potvin. Planned route optimization for real-time vehicle routing. In T.G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 1–18. Springer, 2007.

[15] M.R. Khouadjia, B. Sarasola, E. Alba, E.-G. Talbi, and L. Jourdan. Metaheuristics for dynamic vehicle routing. In *Metaheuristics for Dynamic Optimization*, pages 265–289. Springer, 2013.

[16] A.N. Letchford, S.D. Nasiri, and A. Oukil. Pricing routines for vehicle routing with time windows on road networks. *Computers & Operations Research*, 51:331–337, 2014.

[17] S. Lorini, J.-Y. Potvin, and N. Zufferey. Online vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, 38(7):1086–1090, 2011.

[18] S. Mitrović-Minić, R. Krishnamurti, and G. Laporte. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B*, 38(8):669–685, 2004.

[19] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, and A.V. Donati. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10(4):327–343, 2005.

[20] I. Or. Traveling salesman-type combinatorial problems and their relation to the logistics of blood banking. *PhD thesis, Department of Industrial Engineering and Management Science, Northwestern University, U.S.A.*, 1976.

[21] V. Pillac, M. Gendreau, C. Guéret, and A.L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[22] J.-Y. Potvin, Y. Xu, and I. Benyahia. Vehicle routing and scheduling with dynamic travel times. *Computers & Operations Research*, 33(4):1129–1137, 2006.

[23] H.N. Psaraftis. Dynamic vehicle routing problems. In B.L. Golden and A.A. Assad, editors, *Vehicle routing: Methods and Studies*. North-Holland, 1988.

[24] H.N. Psaraftis, M. Wen, and C.A. Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31, 2016.

[25] V. Pureza and G. Laporte. Waiting and buffering strategies for the dynamic pickup and delivery problem with time windows. *INFOR: Information Systems and Operational Research*, 46(3):165–175, 2008.

[26] J. Respen, N. Zufferey, and J.-Y. Potvin. Impact of online tracking on a vehicle routing problem with dynamic travel times. *RAIRO Operations Research*, 53(2):401–414, 2019.

[27] E. Taniguchi and H. Shimamoto. Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Transportation Research Part C*, 12(3-4):235–250, 2004.

[28] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. SIAM, 2014.

[29] T. Van Woensel, L. Kerbache, H. Peremans, and N. Vandaele. Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research*, 186(3):990–1007, 2008.