CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

The Time-Dependent Location-Routing Problem

Carise E. Schmidt
Arinei C.L. Silva
Maryam Darvish
Leandro C. Coelho

December 2018

CIRRELT-2018-54

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2018-022.

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

The Time-Dependent Location-Routing Problem

Carise E. Schmidt[1], Arinei C.L. Silva[1], Maryam Darvish[2,*], Leandro C. Coelho[2]

[1] GTAO, Universidade Federal do Paraná, Rua XV de Novembro, 1299, CEP 80.060-000, Centro Curitiba, PR Brasil Curitiba, Brazil

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, 2325 de la Terrasse, Université Laval, Québec, Canada G1V 0A6

Abstract. In this paper, we study an important logistics and transportation problem involving facility location and time-dependent vehicle routing. We introduce, model, and solve the time-dependent location-routing problem in which the objective is to select a facility among a set of potential locations in order to minimize the total distribution time, as motivated and necessary in urban contexts. We propose a mathematical formulation and a set of valid inequalities for the problem. We use heuristics to provide good quality initial solutions for the exact method. Moreover, we propose a matheuristic in which different solutions obtained by the heuristics are combined and improved by solving a set covering problem. Finally, we propose a large set of benchmark instances based on real data obtained from logistics operators allowing us to mimic traffic, location, and service time parameters. We conduct extensive computational experiments on these realistic instances assessing the quality of our heuristic, the effectiveness of the valid inequalities in the mathematical model, and the overall matheuristic framework. Important insights and analysis pertaining to city logistics and traffic conditions are provided based on the numerical tests which contain realistic information of up to 5 depot locations, 100 delivery customers, and 15 vehicles, considering traffic patterns that change hourly.

Keywords. Urban logistics, integrated logistics, vehicle routing, location, time-dependent.

Acknowledgements. This project was partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764. This support is greatly acknowledged. We thank Calcul Québec for providing high performance parallel computing facilities, and Hamza Heni and Khaled Belhassine for their help on creating the instances.

---

* Corresponding author: Maryam.Darvish@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2018

1. Introduction

The urban freight transport industry is rapidly growing due to the role it plays in the economic vitality of the cities. With rapid population increase in the urban areas, growth of road freight transportation has been inevitable. The way to route large trucks around and within urban centers is highly important due to its immense social, environmental, and economic impacts. From the research point of view, this has led to the emergence of the green vehicle routing problem (Erdoğan and Miller-Hooks, 2012; Leggieri and Haouari, 2017) and from the practical one, we observe that recently, many cities have invested on improving their transportation systems (Yuan and Yu, 2018). However, at the same time that the urban freight transport is known to be responsible for congestion and pollution in the cities, it is itself a victim of logistic decisions. With cost minimization as the main focus in freight transportation planning (Demir et al., 2014), designing an efficient distribution network in which the least time is spent on the road network, by avoiding the traffic jam, has been of less concern.

In network design problems, one decides on the location of the facilities such as distribution routes (Bektaş et al., 2017). When designing a sustainable urban freight transport network, these decisions become the important factors affecting and being affected by the traffic congestion. However, location and distribution decisions have long been studied separately from one another in the literature and later on models that integrate them are oversimplified and do not reflect the real world. One of these oversimplifications is with respect to the time dependency of the travel time and its effect on the facility location decisions. In most of the models presented in the literature, the objective lies in minimizing the cost of opening a facility and the distribution costs from this facility towards the customers. However, when it comes to the urban freight transport, the travel time changes dynamically throughout the day, consequently affecting operational costs. Given that the trend is to locate facilities far from the city centers and mainly in the periphery of urban areas (Koç et al., 2016), one could argue that in the long run the difference between the cost of potential locations is expected to be similar, whereas the traffic pattern around them can get significantly complicated. The traffic condition does not improve overtime, and while some non-congested areas would become congested, the congested ones remain mostly unchanged. This suggests that in real world the travel time for distribution is not constant and invariant to time. However, in most studies on the integration of location and routing, the traveling time is considered to be constant.

A notable exception is the work of Hu et al. (2018) in which traffic and the number of trucks on the route are considered dependent on one another. We, on the other hand, have real-life information regarding traffic and make tactical and operational decisions geared towards avoiding the existing traffic congestion. Hence, aiming to bridge the gap between two popular research topics, we extend the class of problems that integrate vehicle routing and location decisions to include problems in which the travel time is not constant.

The problem at hand consists of the integration of location decisions with the time-dependent vehicle routing problem (TD-VRP). It is long known that decisions on the location of the facilities and the routing of the vehicles are highly interdependent (Nagy and Salhi, 2007; Drexl and Schneider, 2014). The idea of integrating the facility location problem with the VRP has emerged in the literature when the influence of the transportation cost on the location decisions was first pointed out (Von Boventer, 1961; Maranzana, 1964). Integrating these two decisions has given rise to the emergence of the location-routing problem (LRP). Drexl and Schneider (2017) define a standard LRP as a single objective function problem characterized by deterministic data, only one planning period, discrete locations, no intermediate locations or inventory considerations, single visit from the potential facility and the vehicle to each customer. The problem we study in this paper meets all these characteristics of a standard LRP except that we do not consider location cost factors, as already motivated, focusing on operational costs due to dynamically evolving travel times. Hammad et al. (2017) explore a similar problem but focus on finding system-wide objectives, while our focus is on improving an individual performance (such as a big company) when facing such challenges.

In the context of the VRP, the interest in the TD-VRP is increasing since the work of Malandraki and Daskin (1992) who identify two sources of fluctuations on travel times: temporal variation of traffic and random variation (due to weather condition, accidents, etc.). They argue that the temporal variation can be represented by a deterministic model in which the travel time is defined based on the distance between the nodes and the time of the day. It is just recent that the scientific literature has recognized that these models are of little importance without real data to support the decisions (Ichoua et al., 2003). Ichoua et al. (2003) use the time-dependent speed model with time windows that satisfies the no passing (FIFO) property. They apply a parallel tabu search heuristic to solve static and dynamic problems (see Gendreau et al. (2015) for a comprehensive survey on travel time modeling and solution methods). Recently, Mancini (2017) proposes a two-phase heuristic approach to solve a TD-VRP. In the first phase, a multi-start random constructive heuristic is applied and the solution obtained is used as columns for a set partitioning based formulation. Veenstra and Coelho (2017) formulate an integrated time-dependent shortest path and VRP. They test the formulation on instances generated from real data obtained from Québec City, Canada. Their results show the importance of considering traffic congestion when optimizing delivery routes. Alvarez et al. (2018) incorporate real-time congestion costs into VRP. They compare the results obtained by four different objective functions of minimizing the Euclidean distance, real distance, real time with static congestion, and real time with dynamic congestion. Their results also highlight congestion costs as an important factor in optimizing distribution. The use of traffic information to drive new ways of performing deliveries affect not only operational routing decisions, but also how to offer possible delivery times to the customers and to access parts of the city (Akyol and De Koster, 2018). Finally, Behnke and Kirschstein (2017) show that using alternative paths to avoid traffic can lead to significant greenhouse gas emission reductions. A similar approach was used by Marufuzzaman and Ekşioğlu (2017) who measure their gains in financial terms, also with positive findings.

To the best of our knowledge, the problem we introduce, formulate, and solve in this paper has not yet been studied in the literature. The contributions of this paper are multi-fold. First, we present a mathematical formulation with several valid inequalities (VIs) for this rich and difficult problem; second we develop a matheuristic to efficiently solve the problem. Finally, we analyze the performance of the proposed methods through extensive computational experiments on a set of instances inspired by the real condition of the traffic congestion based on data from Québec City.

The paper is organized as follows. Section 2 presents the formal problem description followed by its mathematical formulation. Section 3 provides an in-depth explanation of the heuristic approach used to solve the problem. Section 4 presents our extensive computational results and discussion. Finally, conclusions and remarks are provided in Section 5.

2. Problem description and formulation

In this section, we formally describe the TD-LRP and present its mathematical formulation. The TD-LRP is defined on a directed and time-dependent graph $G(N, A, H)$, where N represents the node set, A is the set of arcs, and H is the set of time intervals. We define an interval as a period of time over which traffic pattern is constant. Let N_d be the set of all the potential depots and N_c the set of customers. We also consider a set of dummy nodes, called terminals, to be used by each vehicle as they return to the depot, denoted by N_t. Therefore, $N = N_d \cup N_c \cup N_t$.

We define K as the set of $|K|$ homogeneous vehicles each having a limited capacity Q and $\delta(i)$ as a set with $|K|$ identical vehicles for each potential depot $i \in N_d$. Let A_{dc}, A_{cc}, A_{cd} be arc sets such that each arc (i, j) is given from the cartesian products as $A_{dc} = N_d \times N_c$; $A_{cc} = N_c \times N_c$, $i \neq j$, $A_{ct} = N_c \times \delta(i)_{i \in N_d}$ and $A_{td} = \delta(i) \times \{i\}, i \in N_d$, such that $A = A_{dc} \cup A_{cc} \cup A_{dc} \cup A_{td}$.

Our planning horizon is a day which we divide into equal time intervals of $\overline{T}$ units; each time interval $h \in H = \{0, 1, .., h, .., m\}$ where $m + 1$ is the number of time intervals per day. Therefore, $[h\overline{T}, (h + 1)\overline{T} - \varepsilon]$ represents the time interval associated to h and ε is a positive number that indicates the smallest time unit, say, a second. Respectively, the travel time for arc $(i, j) \in A$ in interval h is given by t^h_{ij}.

The demand of each customer $i \in N_c$ is denoted by q_i and its service time by s_i. Knowing that all customers must be served, the objective is to minimize the total travel time. To this end, the best location for a single depot is to be identified along with time-dependent routes to serve the customers. We define the following variables:

- continuous variables a_i represent the departure time from node $i \in N$;
- binary variables y^h_i indicate whether a route leaves from $i \in N$ in time interval $h \in H$;
- binary variables w_i represent whether depot i is selected;
- continuous variable u_i indicate the load on the truck upon departing from customer i;
- binary variables x^h_{ij} represent whether arc (i, j) is traversed by a vehicle in time interval h;
- binary variables z_{ij} equal to 1 if arc (i, j) is traversed by a vehicle and 0 otherwise.

Table 1 summarizes the notation used in our model.

Table 1: Notation used in the model

Parameters	
$m+1$	Number of time interval in a day
$\overline{T}$	The length of each time interval
q_i	Demand of each customer $i \in N_c$
Q	Vehicle capacity
s_i	Service time of each customer $i \in N_c$
t_{ij}^h	Travel time of arc $(i,j) \in A$ in time interval h
Sets	
N_c	Set of customers
N_d	Set of potential depots
N_t	Set of terminals
K	Set of vehicles
H	Set of time intervals
A	Set of arcs
$\delta(i)$	Set with $\lvert K \rvert$ identical vehicles for each potential depot $i \in N_d$
Variables	
a_i	Departure time from node $i \in N$
u_i	Load of the truck when departing from customer i
w_j	If depot j is selected
x_{ij}^h	If arc (i,j) is traversed by a vehicle in time period h
y_i^h	If node $i \in N$ is left at time interval $h \in H$
z_{ij}	If arc (i,j) is traversed by a vehicle
Indices	
h	Time interval
i, j	Nodes

The mathematical formulation is as follows:

$$\min \sum_{(i,j)\in A} \sum_{h\in H} t_{ij}^h x_{ij}^h \tag{1}$$

subject to:

$$\sum_{i\in(N_c\setminus\{j\})\cup N_d} z_{ij} = 1, \quad \forall j \in N_c \tag{2}$$

$$\sum_{j\in(N_c\setminus\{i\})\cup N_t} z_{ij} = 1, \quad \forall i \in N_c \tag{3}$$

$$\sum_{j\in N_c} z_{ij} \leq |K| w_i, \quad \forall i \in N_d \tag{4}$$

$$\sum_{i\in N_c} z_{iv} \leq w_j, \quad \forall v \in \delta(j), \forall j \in N_d \tag{5}$$

$$\sum_{j\in N_c} z_{ij} = \sum_{v\in\delta(i)} z_{vi}, \quad \forall i \in N_d \tag{6}$$

$$\sum_{i \in N_d} w_i = 1 \tag{7}$$

$$u_i - u_j + Qz_{ij} \leq Q - q_j, \quad \forall i, j \in N_c, i \neq j \tag{8}$$

$$q_i \leq u_i \leq Q, \quad \forall i \in N_c \tag{9}$$

$$\sum_{j \in N_c} x^0_{ij} = \sum_{j \in N_c} \sum_{v \in \delta(i)} z_{jv}, \quad \forall i \in N_d \tag{10}$$

$$\sum_{h \in H} x^h_{ij} = z_{ij}, \quad \forall (i, j) \in A \tag{11}$$

$$x^h_{ij} \leq y^h_i, \quad \forall (i, j) \in A, \forall h \in H \tag{12}$$

$$\sum_{h \in H} y^h_i = w_i, \quad \forall i \in N_d \tag{13}$$

$$\sum_{h \in H} y^h_i = 1, \quad \forall i \in N_c \tag{14}$$

$$\sum_{h \in H} y^h_v \leq w_i, \quad \forall v \in \delta(i), \forall i \in N_d \tag{15}$$

$$\sum_{v \in \delta(i)} \sum_{h \in H} y^h_v = \sum_{j \in N_c} \sum_{v \in \delta(i)} z_{jv}, \quad \forall i \in N_d \tag{16}$$

$$a_j = 0, \quad \forall j \in N_d \tag{17}$$

$$a_j \geq a_i + s_j + t^h_{ij} - 2\overline{T}|H|(1 - x^h_{ij}), \quad \forall (i, j) \in A \setminus A_{td}, \forall h \in H \tag{18}$$

$$a_j \leq a_i + s_j + t^h_{ij} + \overline{T}|H|(1 - x^h_{ij}), \quad \forall (i, j) \in A \setminus A_{td}, \forall h \in H \tag{19}$$

$$\sum_{h \in H} h\overline{T}y^h_i \leq a_i \leq \sum_{h \in H} h\overline{T}y^h_i + \overline{T} - \varepsilon, \quad i \in N_c \tag{20}$$

$$\sum_{h \in H} h\overline{T}y^h_v \leq a_v \leq \sum_{h \in H} h\overline{T}y^h_v + \sum_{h \in H} y^h_v(\overline{T} - \varepsilon), \quad \forall v \in \delta(i), \forall i \in N_d \tag{21}$$

$$w_i \in \{0, 1\}, \quad \forall i \in N_d \tag{22}$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A \tag{23}$$

$$y^h_i \in \{0, 1\}, \quad \forall i \in N, \forall h \in H \tag{24}$$

$$x^h_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A, \forall h \in H \tag{25}$$

$$a_i \in \mathbb{R}_+, \quad \forall j \in N \tag{26}$$

$$u_i \in \mathbb{R}_+, \quad \forall i \in N_c. \tag{27}$$

The objective function (1) minimizes the total driving time. Constraints (2) assure that each customer is visited exactly once, either from another customer or a depot. Similarly, constraints (3) guarantee that after vising each customer, the vehicle will either visit another customer or goes back to a depot. Constraints (4) impose that the number of arcs leaving a selected depot is at most equal to the fleet size. Each terminal node associated with a depot is visited at most once, if and only if the depot is selected, as ensured by constraints (5). Constraints (6) ensure that the number of vehicles leaving a depot is equal to the number of vehicles entering the terminal nodes associated with that depot. Note that one terminal node is associated with each vehicle, and all vehicles return to the depot node. Obviously, the distances and travel times from the terminal nodes to the depot is zero. Constraints (7) ensure that only one depot is selected. Originally proposed by Kulkarni and Bhave (1985), constraints (8) and (9) are the extension of the Miller-Tucker-Zemlin subtour elimination. Constraints (10) guarantee that the vehicles start their routes in the first period, and that the same number of vehicles return to the terminal nodes. Each arc (i, j) is visited in a single time interval as imposed by constraints (11). Constraints (12) enforce that if an arc (i, j) is traversed by a vehicle in time interval h, then h is the time interval considered in the departure from the origin i. Constraints (13) guarantee a departure from a depot only if the depot is selected. Similarly, constraints (14) guarantee that only one time interval h is associated with the departure from customer i, and constraints (15) indicate that the vehicle can only access the terminal of the selected depot. Constraints (16) guarantee that a terminal is visited only if there exists an arc from a customer linking to it. Constraints (17) set the departure time from all depots to zero. Using constraints (18) and (19), we control the departure time from the nodes. The departure time to node j from node i includes the departure time to node i, the time it takes to traverse arc (i, j), and the service time at node j. The departure time from each node i is linked to subsequent time intervals as shown by (20) and (21). Finally constraints (22)–(27) enforce integrality and non-negativity conditions on the variables.

Note that constraints (11), (12), (23) and (24) force the x variables to belong to the interval $[0, 1]$. Moreover, considering constraints (13)–(15) and (22), it is possible to show that, even relaxing the integrality requirement on the x variables, a solution with the same cost/time exists in which all x variables remain binary. Thus, constraints (25) shall be relaxed, and the x variables can be considered as continuous ones.

We also define the following VIs to strengthen the model.

Constraints (8) can be lifted as in Kara et al. (2004), yielding (28):

$$u_i - u_j + Qz_{ij} + (Q - q_i - q_j)z_{ji} \leq Q - q_j, \quad \forall i, j \in N_c, i \neq j. \tag{28}$$

As imposed by constraints (17), all departures from the depot happen at $h = 0$, therefore, we can easily reduce the size of the problem by removing some of the variables:

$$y_i^h = 0, \quad \forall i \in N_d, \forall h \in H \setminus \{0\} \tag{29}$$

$$x_{ij}^h = 0, \quad \forall i \in N_d, \forall j \in N_c, \forall h \in H \setminus \{0\}. \tag{30}$$

In the first time interval, some variables associated with the departure from customers can also be removed from the problem. This is the case when the service time of a customer plus the shortest time to traverse the arc between a depot and this customer is greater than the length of the time interval:

$$x_{ij}^0 = 0 \quad \forall i \in N_c \mid \left(\min_{a \in (N_c \setminus \{i\}) \cup N_d} \{t_{ai}^0\} + s_i \right) \geq \overline{T}, \quad \forall j \in N_c \cup N_t, j \neq i \tag{31}$$

$$x_{ij}^0 = 0 \quad \forall i \in N_c \mid \left(\min_{\substack{a \in N_d \\ b \in N_c}} \{t_{ab}^0\} + s_i \right) \geq \overline{T}, \quad \forall j \in N_c \cup N_t, j \neq i. \tag{32}$$

Similarly, for the last time interval, it is also possible to remove some variables associated with the terminal nodes from the problem. This is the case when the time to traverse an arc and to arrive at a terminal is greater than the length of the time interval, or if the shortest time to leave from a customer toward a terminal, plus the service time is greater than the length of the time interval, or yet if the sum of the shortest time to arrive at a customer, its service time, and the shortest time to leave from the customer to a terminal is greater than the length of the time interval.

$$x_{ij}^m = 0 \quad \forall i \in N_c \mid \min_{a \in N_t} \{t_{ia}^m\} \geq \overline{T}, \quad \forall j \in N_t \tag{33}$$

$$x_{ij}^m = 0, \quad \forall j \in N_c \mid \left(\min_{a \in N_c \setminus \{j\}} \{t_{aj}^m\} + s_j + \min_{b \in N_t} \{t_{jb}^m\} \right) \geq \overline{T}, \quad \forall i \in N_c, i \neq j \tag{34}$$

$$x_{ij}^m = 0, \quad \forall i \in N_c, \forall j \in N_t \mid t_{ij}^m \geq \overline{T}. \tag{35}$$

We can establish a lower bound as the sum of several shortest times. Let k' be the minimum number of vehicles required to meet the customer demand considering a vehicle capacity Q, where k' is obtained as a solution of a bin packing problem. Let $f_v = \{\min_{a \in N_c} \{t_{va}^0\} + \min_{\substack{b \in N_c \\ c \in \delta(v) \\ h \in H}} \{t_{bc}^h\}\}, \forall v \in N_d$ and $f_{v_n} \mid f_{v_1} \leq f_{v_2} \leq \ldots \leq f_{v_{|N_d|}}$. Similarly, let $g_v = \{\min_{\substack{a \in N_c \setminus \{v\} \\ h \in H}} \{t_{va}^h\}\}, \forall v \in N_c$ and $g_{v_n} \mid g_{v_1} \leq g_{v_2} \leq \ldots \leq g_{v_{|N_d|}}$.

$$\sum_{(i,j) \in A \setminus A_{td}} \sum_{h \in H} t_{ij}^h x_{ij}^h \geq \sum_{n=1}^{k'} f_{v_n} + \sum_{n=1}^{|N_c|-k'} g_{v_n}. \tag{36}$$

We can also improve the routing part of the model by forbidding subtours of sizes two and three:

$$z_{ij} + z_{ji} \leq 1, \quad \forall i, j \in N_c, i \neq j \tag{37}$$

$$z_{ij} + z_{ji} + z_{iv} + z_{vi} + z_{jv} + z_{vj} \leq 2, \quad \forall i, j, v \in N_c, i \neq j, \neq v. \tag{38}$$

Constraints (39) set the number of vehicles leaving from a depot equal to number of vehicles arriving at the terminals associated with this depot. Note that in the TD-LRP a single depot is selected.

$$\sum_{j \in N_c} z_{ij} = \sum_{v \in \delta(i)} \sum_{j \in N_c} z_{jv}, \quad \forall i \in N_d. \tag{39}$$

The outbound flows from the depot is guaranteed by inequalities (40) and (41). The result of a bin packing problem determines the minimum number of vehicles which is set as the lower bound for all departures from the selected depot and also the total number of departures from all depots.

$$\sum_{j \in N_c} x_{ij}^0 \geq w_i k', \quad \forall i \in N_d \tag{40}$$

$$\sum_{i \in N_d} \sum_{j \in N_c} z_{ij} \geq k'. \tag{41}$$

The same time interval has to be used in the solution for both variables y_i^h and x_{ij}^h:

$$\sum_{j \in N_c \setminus \{i\} \cup N_t} x_{ij}^h = y_i^h, \quad \forall i \in N_c, \forall h \in H. \tag{42}$$

Finally, we can avoid some symmetric solutions by imposing an order on the use of the terminal nodes:

$$\sum_{j \in N_c} z_{jv} \geq w_i, \quad \forall v \in \delta(i) \mid 1 \leq v \leq k', \quad \forall i \in N_d \tag{43}$$

$$a_v \geq a_{v+1}, \quad \forall v \in \delta(i) \mid 1 \leq v \leq k' \wedge v < |K|, \quad \forall i \in N_d. \tag{44}$$

3. Matheuristic algorithm

In this section, the general framework of our heuristic approach is explained in detail. We first create a set of different initial solutions and build a diverse set of individual vehicle routes that do not constitute a full solution. This is conducted by applying fast constructive heuristics presented in Section 3.1. Non-dominated routes are processed and used as potential variables for a route-based set covering model described in Section 3.2. We also use a set containing the best solutions obtained from the procedures described in Section 3.1 as initial solutions to warm start the model presented in Section 2. These experiments are described in Section 4.

3.1. Constructive heuristics

We now describe seven constructive procedures used to create vehicle routes. If a solution is built such that all customers are visited, it is added to a set Λ of feasible solutions. If the constructive heuristic fails to provide a feasible solution, its partial feasible routes are added to the set λ of partial solutions, containing only feasible routes.

Given the random aspect of these constructive procedures, they are repeated several times until a stopping criterion is met. As show in Algorithm 1, the execution of each heuristic is repeated until the total number of solutions added to either set Λ or λ reaches a threshold, L_1 for full solutions and L_2 for partial ones.

Our proposed constructive heuristics for the TD-LRP are described next.

1. **Random route construction:** in this heuristic we start with an empty list for each vehicle from each depot, and another list for all unvisited customers. We randomly select a customer

Algorithm 1 Stopping criteria for the constructive heuristic

Input: a constructive heuristic procedure, limits of full solutions (L_1) and partial solutions (L_2) obtained
Output: up to L_1 solutions or L_2 partial solutions using the input heuristic
for each depot **do**
  Number of iterations $l_1 = l_2 = 0$,
  while $l_1 < L_1$ **and** $l_2 < L_2$ **do**
    Initialize $|K|$ empty routes, and set $[j]$ as the list of all (non-visited) customers
    Apply the constructive heuristic
    if full solution obtained **then**
      $l_1 \leftarrow l_1 + 1$
    else
      $l_2 \leftarrow l_2 + 1$
    end if
  end while
end for

from the latter and add it to the end of the route of the first vehicle, if feasible. If this route is no longer feasible, the customer is assigned to a new route. Once all customers are successfully assigned to a route, this full solution is added to the set Λ. Otherwise, in case some customers are left unvisited, partially generated routes are added to the set λ. The procedure stops when the number of elements added to either set Λ or λ reaches a threshold. The pseudocode for this procedure is described in Algorithm 2. Each feasible solution for a depot is also replicated for all other depots and if a feasible solution is obtained, it is added to set Λ, otherwise the last customers on the infeasible routes are removed until the individual route becomes feasible. Then the partial solution is added to the set λ. Each generated solution is also replicated for every depot as described in Algorithm 3.

Algorithm 2 Random route construction

loop
  Select a (non-visited) customer from $[j]$ and insert the customer at the end of the current route
  if the route is feasible **then**
    Remove customer from $[j]$
    if $[j]$ is empty **then**
      Apply Algorithm 3 to replicate the solution to all other depots
      Return a full solution for set Λ
    end if
  else
    Remove the last customer from the route
    if there is a vehicle available **then**
      Start a new current route with the removed customer.
    else
      Return a partial solution for set λ
    end if
  end if
end loop

Algorithm 3 Depot replication

```
Input: a feasible solution from depot d
Output: new solutions (potentially infeasible) for all other depots with the same sequence of customers
for all v ∈ N_d, v ≠ d do
    for every route of the solution do
        Change depot d for v
        while the new route is infeasible do
            Randomly remove one customer
        end while
    end for
    if the new solution is feasible then
        Return a full solution for set Λ
    else
        Return a partial solution for set λ
    end if
end for
```

2. **Random parallel route construction:** a drawback of the previous route creation process is that we start by assigning as many customers as we can to the first vehicles, which causes only a few customers remain to be assigned to the last vehicles. This issue is overcome in this heuristic by starting all routes simultaneously. Customers are sequentially assigned to the vehicles in an alternate manner. This construction heuristic is described in Algorithm 4. We repeat this heuristic until the same stopping criteria explained in Algorithm 1 is met. As before, each feasible solution is replicated for all other depots, as in Algorithm 3.

Algorithm 4 Random parallel route construction

```
loop
    while [j] is not empty do
        for each vehicle do
            Select a (non-visited) customer from [j]
            Insert the customer at the end of the current route
            Remove customer from [j]
        end for
    end while
    if the solution is feasible then
        Return a full solution for set Λ
    else
        while a route is infeasible do
            Remove one customer from the route
        end while
        Return a partial solution for set λ
    end if
end loop
```

3. **ν-closest neighbors:** in this heuristics, the search starts from a depot for which we identify its ν closest customers; a random neighbor customer is selected and added to the first route,

leaving the depot in $h = 0$. The travel and service times are then computed and the *ready time* is identified – the time at which the vehicle is ready to leave the newly added customer. We continue applying this same procedure on each customer added to the route and now its closest unvisited neighbors. Note that the notion of "a close neighbor" is based on the ready time of the latest added customer. Here, we need to ensure that addition of the customer to the route is feasible with respect to the vehicle capacity and available time. If the route is feasible, the procedure continues; otherwise, the selected customer returns to the list of unvisited customers, a new vehicle is selected, and the procedure restarts with another unvisited neighbor of the depot. Again, if the final solution is feasible, i.e., all customers are successfully added to a route, this solution is added to set Λ. Otherwise, it is deemed infeasible and its partial routes are saved by adding the solution to the set λ.

4. **ν-closest neighbors – backward:** this method is similar to the previous one, except that we start assigning the customers from the last time interval to the first. The motivation is to choose the best customers towards the end of the route with respect to the driving time. Once the route is created, its sequence is then evaluated by departing from the depot at time zero. Algorithm 5 presents the pseudocode for the ν-closest neighbors and its backward version.

5. **Parallel ν-closest neighbors:** this heuristic is similar to the *ν-closest neighbors*, except that several vehicles are simultaneously selected at each iteration. Obviously, all vehicles start their routes at time $h = 0$ from the depot heading to an unvisited neighbor.

6. **Parallel ν-closest neighbors – backward:** as in *Parallel ν-closest neighbors*, here we create routes from the end of time horizon towards the beginning, but start all routes in parallel, adding one customer to each route at each iteration. Once all the customers are added to the routes, we evaluate the value of the objective function, in a forward manner. The pseudocode for these last two heuristics is presented in Algorithm 6.

7. **Route enumeration:** in this heuristic, we create optimal routes consisting of up to γ customers. The motivation is to have short routes computed optimally and added to the set Λ. The pseudocode of this procedure is described in Algorithm 7.

All routes added to sets Λ and λ are checked against the already added ones, such that different sequences of the same set of customers appears only once; the value of the objective function is

Algorithm 5 ν-closest neighbors

```
if Forward then
   For the customer on the list, right before the depot node at the end:
   order the travel times from the node to all not visited customers j
end if
if Backward then
   For the customer on the list, right after the depot node in the beginning:
   order the travel times from the node to all not visited customers j
end if
Randomly draw a customer among the k not visited closet neighbors j considering the time interval
if Forward then
   Insert the customer j after the last node just before the end depot i in the current routing list
end if
if Backward then
   Insert the customer j before the last node just after the beginning depot i in the current routing list
end if
if The route is feasible then
   Remove the customer from [j]
   if [j] is empty then
      Return a full solution for set Λ
   end if
else
   Remove one customer j from the current routing list
   if There is vehicle available then
      Start a new route. Add i in the beginning and in the end of the route
   else
      Return a partial solution for set λ
   end if
end if
```

Algorithm 6 Parallel ν-closest neighbors - Forward/Backward

```
if Forward then
   For the customer on the list, right before the depot node at the end:
   order the travel times from the node to all not visited customers j
end if
if Backward then
   For the customer on the list, right after the depot node in the beginning:
   order the travel times from the node to all not visited customers j
end if
Randomly draw a customer among the k not visited closet neighbors j considering the time interval
if Forward then
   Insert the customer j after the last node just before the end depot i in the current routing list
end if
if Backward then
   Insert the customer j before the last node just after the beginning depot i in the current routing list
end if
if [j] is empty then
   Exit loop
end if
if the solution is feasible then
   Return a full solution for set Λ
else
   while a route is infeasible do
      Remove one customer from the route
   end while
   Return a partial solution for set λ
end if
```

Algorithm 7 Route enumeration of up to γ customers

```
for each i ∈ N_d do
   for i = j to N_c do
      DepotArrivalTime ← BigNumber
      for each possible arrangement of γ customers j ∈ N_c do
         Insert the depot i in the beginning and at the end of each combination with γ customers
         if This route is feasible then
            Calculate the accumulated travel time on this route
            if The accumulated travel time is less than DepotArrivalTime then
               DepotArrivalTime ← Accumulated travel time of the current route
               Best route ← Current route
            end if
         end if
         Add the best route to set Λ
      end for
   end for
end for
```

updated if a better sequence is found.

Moreover, in order to further diversify the search, we execute heuristics 1–6 with fewer time intervals. The goal is to to obtain relatively shorter routes, therefore, they are executed by reducing the number of periods by one and two, and all solutions (feasible or not) are added to sets Λ and λ.

3.2. Set covering model

The last phase of our matheuristic is to try to combine different routes from sets Λ and λ to obtain better solutions. This is done using a set covering model as follows. The mathematical formulation and details on the problem are as follows.

Given a set of n elements (customers) N_c, and a set of vehicle routes R whose union equals N_c. Each route r in R has an associated cost c_r which represents the accumulated time to execute the route and an associated parameter Ω_r indicating the depot linked to route r. We use α_{rj} as the assignment of customer j in the set r, that is, $\alpha_{jr} = 1$ if customer j is visited by route r, and 0 otherwise. K is the set of vehicles for each depot.

Two sets of variables are defined. For each route $r \in R$, binary variable χ_r takes value 1 if the set of customers r is visited, and 0 otherwise. For each depot $d \in N_d$, binary variable Υ_d takes value 1 if depot d is used, 0 otherwise.

The formulation is the following:

$$\min \sum_{i \in R} c_r \chi_r \tag{45}$$

subject to

$$\sum_{r \in R} \alpha_{rj} \chi_r = 1, \quad \forall j \in N_c \tag{46}$$

$$\sum_{r \in R} \chi_r \leq |K| \tag{47}$$

$$\sum_{d \in N_d} \Upsilon_d = 1 \tag{48}$$

$$\chi_r \leq \Upsilon_{(\Omega_r)}, \quad \forall \Omega_r \in N_d, \forall r \in R \tag{49}$$

$$\chi_r \in \{0, 1\}, \quad \forall r \in R \tag{50}$$

$$\Upsilon_d \in \{0,1\}, \quad \forall d \in N_d. \tag{51}$$

The objective function (45) is to find the subset of routes with the minimum total execution time. Each customer has to be visited exactly once by constraints (46). Constraint (47) guarantees that the number of routes used does not exceed the number of vehicles available. By constraint (48) we assure that all the used routes are associated with a single depot. Constraints (49) enforce the connection between each route and its respective depot. Constraints (50) and (51) define the nature and domain of the variables.

4. Computational experiments

In this section, we provide details on our instances, the setting and parameters of our algorithms and extensive results along with an elaborated analysis. The algorithms are coded in C++ and we use Gurobi Optimizer 8.0.1 as the MIP solver in its default settings. All computational experiments are conducted on an Intel Core i7 processor running at 3.4 GHz with 64 GB of RAM installed, with the Ubuntu Linux operating system. Four threads are used and a total time limit of 10800 seconds is imposed for each execution. Section 4.1 describes how the instances are generated and the results of detailed computational experiments are provided in Section 4.2.

4.1. Instance generation

To conduct our computational experiments, we have randomly generated several classes of instances. Exploiting the same database presented in Belhassine et al. (2018), geographical information available from the real road network and traffic of the Québec City is used to generate our instances. The following parameters are generated based on the data from an industrial partner working in the retailing of furniture and appliances.

Over a planning horizon of 15 hours, from 6:00 to 21:00, we consider three equal-length intervals of 3600, 5400, and 10800 seconds which consequently differentiate between our *large*, *medium*, or *small* instances. We consider 1, 3, or 5 available depots for each instance in which 10, 20, 50, 80, or 100 customers are located. The demand of each customer is randomly generated from $[50, 750]$ units and the service time varies between $[1000, 10800]$ seconds. Therefore, each instance is identified by the number of time intervals ($|H|$), depots ($|N_d|$), customers ($|N_c|$), vehicles ($|K|$), and the capacity of each vehicle (Q) set from preliminary tests.

For each combination of parameters, we generate 5 random instances. Table 2 summarizes the parameters used in our instances.

Table 2: Instance sets and the parameters

Type	$\|H\|$	$\overline{T}$	$\|N_d\|$	$\|N_c\|$	$\|K\|$	Q
small	5	10800	$\{1,3,5\}$	10	3	4000
			$\{1,3,5\}$	20	4	4000
			$\{1,3,5\}$	50	8	4500
			$\{1,3,5\}$	80	12	4500
			$\{1,3,5\}$	100	15	4500
medium	10	5400	$\{1,3,5\}$	10	3	4000
			$\{1,3,5\}$	20	4	4000
			$\{1,3,5\}$	50	8	4500
			$\{1,3,5\}$	80	12	4500
			$\{1,3,5\}$	100	15	4500
large	15	3600	$\{1,3,5\}$	10	3	4000
			$\{1,3,5\}$	20	4	4000
			$\{1,3,5\}$	50	8	4500
			$\{1,3,5\}$	80	12	4500
			$\{1,3,5\}$	100	15	4500

4.2. Computational results

We now present the results of our extensive computational experiments. We have tested several parameters to tune our methods and the following are proven to provide the best average results. For all our heuristics, we use $\nu = 2$, $L_1 = 5000$, and $L_2 = 10000$.

We start our analysis by showing in Section 4.2.1 the results obtained by applying the constructive heuristic presented in the previous section. Then in Section 4.2.2 the results of the mathematical model of Section 2 are presented; once it is fed with a pool of initial solutions from our heuristics and with all valid inequalities added. We then compare these results when no initial solution is provided, and when valid inequalities are removed. Then, in Section 4.2.3 we provide detailed results using our matheuristic of Section 3.

4.2.1. Results of the constructive heuristic

Table 3 shows the results obtained by applying the proposed constructive heuristic. On the first two columns of this table we provide general information on the instance, and then we report the results obtained for potential depots on the instance. For each potential depot, we report the number of vehicle's routes generated by constructive heuristic (# routes generated), followed by the best solution obtained by constructive heuristic (Best solution), and finally the execution time in seconds (Time). We first observe that the number of time intervals in the instance, determining its size (small, medium, large) has no effect on the complexity of the problem, as indicated by the relatively constant number of routes and time to generate them. Also, by exploring more than 20 million routes for some configurations, it is noticeable that our heuristics are capable of providing very diverse routes. Moreover, the provided averages are coherent with the fact that the value of the objective function decreases when the number of potential depots increases, as the base instance is the same but new locations are added for the new potential depot, meaning that the objective function should go down, at the expense of a more difficult optimization problem. Finally, the average execution time is below an hour, which is very acceptable to solve such a challenging tactical optimization problem.

Table 3: Average results from the constructive heuristic

Instance	$\lvert N_c\rvert$	1 depot			3 depots			5 depots		
		# routes generated	Best solution	Time (s)	# routes generated	Best solution	Time (s)	# routes generated	Best solution	Time (s)
Small	10	369078.40	7846.40	15	1226114.40	7439.00	48	2037187.20	7904.20	68
	20	738245.00	14337.40	38	2245948.20	13824.00	115	3806241.80	12773.60	196
	50	2479664.80	29915.40	220	5918252.20	26654.20	591	9894822.60	26377.00	970
	80	4182447.20	44227.80	1160	10059494.80	37466.00	3263	15631462.40	35763.80	5170
	100	5308720.60	42219.60	1638	13027628.40	46539.40	4389	20383988.40	39356.60	7734
Average		2615631.20	27709.32	614	6495487.60	26384.52	1681	10350740.48	24435.04	2828
Medium	10	330806.20	8123.20	14	1009597.40	7512.60	39	1671495.00	7848.40	60
	20	608242.60	14811.60	37	1401006.20	13785.00	92	2625247.00	13113.40	160
	50	2165547.00	31184.20	204	4990314.40	26914.80	531	8029309.20	26475.00	862
	80	3793161.00	45658.20	707	9273097.20	39801.40	2978	14230304.80	35950.20	5115
	100	4452738.00	40396.00	1132	11595049.80	39929.80	4596	17992807.60	34278.80	5828
Average		2270098.96	28034.64	419	5653813.00	25588.72	1647	8909832.72	23533.16	2405
Large	10	332077.20	7920.20	14	995294.60	7467.40	40	1647919.20	7479.80	58
	20	447788.40	14347.00	32	1361307.80	13887.80	89	2379870.00	12972.00	152
	50	1803903.60	29372.20	190	4763696.80	27027.20	522	8121700.00	27053.00	857
	80	3318467.40	44811.40	530	8042322.80	39308.60	1397	12877642.00	37208.20	2281
	100	4379811.00	42183.20	1115	10303364.00	46000.00	4079	16530344.80	38183.20	5124
Average		2056409.52	27726.80	376	5093197.20	26738.20	1225	8311495.20	24579.24	1694
Global average		2314046.56	27823.59	470	5747499.27	26237.15	1518	9190689.47	24182.48	2309

The best solution obtained from the constructive heuristic is used as one initial solution (IS) for both solving the mathematical model and the second phase of our proposed matheuristic.

4.2.2. Results of the mathematical model

In what follows we present the results from our model when it is fed with an initial solution and under the presence of all valid inequalities. Table 4 presents the results for 1 depot, Table 5 for cases with 3 depots, and finally Table 6 for the largest instances containing 5 potential depots. As before, the first two columns of the tables provide information on the instance. In order to compare the performance of these different methods, on each table we provide information on the obtained upper bound (UB) and lower bound (LB), the gap (in %) calculated as $100(UB - LB)/UB$, the execution time (in seconds) and finally, the solution (UB) improvement over the best solution obtained with the constructive heuristic (in %).

Table 4: Average results for instances with 1 depot from the TDLRP with VIs and with initial solutions

Instance	$\|N_c\|$	IS	UB	LB	Gap (%)	Time (s)	UB improvement over IS (%)
Small	10	7846.40	7595.40	7595.40	0.00	12	3.29
	20	14337.40	13450.20	11207.63	16.99	10800	6.10
	50	29915.40	28876.60	16260.00	42.88	10800	3.65
	80	44227.80	43623.20	22251.10	56.46	10801	1.41
	100	42219.60	41391.20	20527.90	50.09	10801	1.79
	Average	27709.32	26987.32	15568.41	33.28	8643	3.25
Medium	10	8123.20	7923.20	7923.20	0.00	145	2.32
	20	14811.60	13749.20	11172.42	18.76	10800	6.79
	50	31184.20	30594.20	16196.92	46.43	10800	1.88
	80	45658.20	45238.60	22809.76	58.71	10801	0.91
	100	40396.00	40267.80	20116.90	51.31	10803	0.29
	Average	28034.64	27554.60	15643.84	35.04	8670	2.44
Large	10	7920.20	7803.00	7803.00	0.00	110	1.44
	20	14347.00	13708.20	11127.29	18.65	10800	4.22
	50	29372.20	28762.00	16118.04	43.26	10801	1.98
	80	44811.40	44463.00	22685.32	58.45	10801	0.82
	100	42183.20	41988.20	20608.22	50.39	10804	0.45
	Average	27726.80	27344.88	15668.37	34.15	8663	1.78
Global average		27823.59	27295.60	15626.87	34.16	8659	2.49

Table 5: Average results for instances with 3 depots from the TDLRP with VIs and with initial solutions

Instance	$\|N_c\|$	IS	UB	LB	Gap (%)	Time (s)	UB improvement over IS (%)
Small	10	7439.00	7030.40	7030.40	0.00	56	5.43
	20	13824.00	13065.00	10868.40	16.42	10800	5.56
	50	26654.20	25988.40	16017.90	38.16	10800	2.57
	80	37466.00	37043.60	19855.88	54.91	10801	1.08
	100	46539.40	46133.40	24032.74	47.81	10802	0.85
	Average	26384.52	25852.16	15561.06	31.46	8652	3.10
Medium	10	7512.60	7252.60	7252.60	0.00	166	3.45
	20	13785.00	13403.60	10675.50	20.24	10800	2.76
	50	26914.80	26194.60	16063.12	38.55	10801	2.72
	80	39801.40	39529.40	20200.26	57.63	10802	0.67
	100	39929.80	39853.00	20319.55	51.76	10804	0.37
	Average	25588.72	25246.64	14902.21	33.63	8675	1.99
Large	10	7467.40	7153.00	7153.00	0.00	153	4.17
	20	13887.80	13630.60	10607.78	22.14	10800	1.79
	50	27027.20	26646.00	15941.90	40.15	10801	1.42
	80	39308.60	39045.80	20120.80	57.33	10802	0.73
	100	46000.00	45941.40	23813.92	47.97	10807	0.12
	Average	26738.20	26483.36	15527.48	33.52	8672	1.64
Global average		26237.15	25860.72	15330.25	32.87	8666	2.24

These tables show that even with all valid inequalities and inputting a set of initial solutions to the model, the gap is still very high after three hours of computing time, suggesting that the problem remains a very difficult optimization problem. It also shows that, typically, Gurobi is not able to

Table 6: Average results for instances with 5 depots from the TDLRP with VIs and with initial solutions

Instance	$\lvert N_c \rvert$	IS	UB	LB	Gap (%)	Time (s)	UB improvement over IS (%)
Small	10	7904.20	7604.00	7604.00	0.00	401	3.65
	20	12773.60	12316.20	9947.64	19.19	10800	3.58
	50	26377.00	25473.00	15735.92	38.23	10800	3.40
	80	35763.80	35108.40	19457.50	51.99	10801	1.83
	100	39356.60	39203.00	19767.86	49.72	10802	0.38
Average		24435.04	23940.92	14502.58	31.83	8721	2.57
Medium	10	7848.40	7788.40	7788.19	0.00	1990	1.00
	20	13113.40	12800.20	10174.50	20.28	10800	2.38
	50	26475.00	25902.80	15727.84	39.23	10800	2.19
	80	35950.20	35722.20	19650.34	53.00	10802	0.63
	100	34278.80	34214.60	16434.26	56.20	10806	0.16
Average		23533.16	23285.64	13955.03	33.74	9040	1.27
Large	10	7479.80	7355.40	7355.37	0.00	714	1.61
	20	12972.00	12793.00	9894.29	22.76	10800	1.38
	50	27053.00	26746.00	15707.58	41.23	10801	1.12
	80	37208.20	37092.60	19396.30	54.04	10803	0.33
	100	38183.20	38031.00	18958.20	51.47	10851	0.47
Average		24579.24	24403.60	14262.35	33.90	8794	0.98
Global average		24182.48	23876.72	14239.99	33.16	8851	1.61

significantly improve the initial solutions provided, with an average improvement of only 2.49% for instances with 1 depot, 2.24% for instances with 3 depots, and 1.61% for instances with 5 depots. Also, for larger instances, the improvements are only marginal, indicating that the use of our initial solution was indeed efficient. In what follows, we evaluate whether the quality of the initial solutions makes a difference and if the valid inequalities are important in this context.

In Table 7 we first analyze the case where initial solutions are not provided, but we keep all valid inequalities present in the model. Here, the solver (Gurobi) relies on applying the branch-and-bound algorithm and its internal heuristics, which are not described in details but known to include at least Minimum Relaxation Heuristic, Feasibility Pump Heuristic, Relaxation induced neighborhood search (RINS) Heuristic, and Zero Objective Heuristic. An asterisk indicates that no solution was obtained for any of the five instances represented by each row.

The results on Table 7 show that for large instances, no feasible solution could be found within the three hours allotted. Columns UB gap (%) and LB gap (%) refer to the deterioration of the solutions (in %) compared to the ones shown in Tables 4–6, and indicate that for the few instances for which a feasible solution was obtained, their quality were significantly worse than those provided by the constructive heuristics (about 4% worse for all number of depots). As for the lower bounds, these were mostly equivalent, with a small (around 0.5%) improvement over the previous case. It is clear that on average the UB worsens and that providing an initial solution has a strong positive impact on the ability of the solver to find feasible solutions. For most instances, the solver was not

Table 7: Average results from the TDLRP with valid inequalities and without initial solutions

Instance	$\|N_c\|$	1 depot UB	1 depot LB	1 depot UB gap (%)	1 depot LB gap (%)	3 depots UB	3 depots LB	3 depots UB gap (%)	3 depots LB gap (%)	5 depots UB	5 depots LB	5 depots UB gap (%)	5 depots LB gap (%)
Small	10	7595.40	7595.40	0.00	0.00	7030.40	7030.40	0.00	0.00	7604.00	7604.00	0.00	0.00
	20	13203.40	11157.46	-1.87	-0.40	13450.20	10790.90	2.69	-0.71	12219.20	9998.21	-0.68	0.47
	50	29487.00†	16184.46	8.93†	-0.45	25329.00‡	15885.58	3.00‡	-0.82	*	15507.67	*	-1.47
	80	*	22176.82	*	-0.34	*	19864.55	*	0.03	*	19422.93	*	-0.18
	100	*	20493.61	*	-0.15	*	24038.71	*	0.02	*	19779.76	*	0.07
Average		*	15521.55	2.36	-0.27	*	15522.03	1.90	-0.30	*	14462.52	-0.34	-0.22
Medium	10	7923.20	7923.20	0.00	0.00	7252.60	7252.44	0.00	0.00	7788.40	7678.90	0.00	-1.10
	20	14943.00	10912.97	7.15	-2.29	14291.40	10488.24	6.02	-1.81	14442.00	9911.01	11.43	-2.64
	50	*	16123.34	*	-0.47	*	16034.77	*	-0.17	*	15655.10	*	-0.47
	80	*	22799.34	*	-0.03	*	20253.26	*	0.25	*	19643.48	*	-0.06
	100	*	20125.01	*	-0.13	*	20354.02	*	0.18	*	16410.72	*	0.18
Average		*	15576.77	3.57	-0.58	*	14876.55	3.01	-0.31	*	13859.84	5.71	-0.82
Large	10	7803.00	7802.95	0.00	0.00	7153.00	7152.94	0.00	0.00	7355.40	7355.35	0.00	0.00
	20	15693.80	10829.14	12.48	-2.81	15774.20	10441.70	13.16	-1.63	15229.00	9719.32	15.01	-1.73
	50	*	16029.47	*	-0.57	*	15867.01	*	-0.48	*	15577.98	*	-0.83
	80	*	22653.03	*	-0.15	*	20115.24	*	-0.04	*	19374.90	*	-0.11
	100	*	20619.60	*	0.04	*	23838.89	*	0.11	*	19024.66	*	0.43
Average		*	15586.84	6.24	-0.70	*	15483.16	6.58	-0.41	*	14210.44	7.50	-0.45
Global average		*	15561.72	4.06	-0.52	*	15293.91	3.83	-0.34	*	14177.60	4.29	-0.50

†One instance (out of five) did not have a feasible solution. The gap is calculated based on the solutions of the four remaining instances.

‡Three instances (out of five) did not have a feasible solution. The gap is calculated based on the solutions of the two remaining instances.

able to obtain any feasible solutions without inputting ours. Moreover, the quality of the solutions obtained are significantly worse than the ones obtained by our constructive heuristics. Moreover, the detailed results (available upon request) reveal that the time it takes to prove optimality for the small instances is decreased by 4.88% when initial solutions are provided. Therefore, it is evident that the results presented in Tables 4–6 rely strongly on the quality of the provided initial solutions.

Table 8: Average results from the TDLRP without valid inequalities and with initial solutions

Instance	$\|N_c\|$	1 depot UB	1 depot LB	1 depot UB gap (%)	1 depot LB gap (%)	3 depots UB	3 depots LB	3 depots UB gap (%)	3 depots LB gap (%)	5 depots UB	5 depots LB	5 depots UB gap (%)	5 depots LB gap (%)
Small	10	7595.40	7595.40	0.00	0.00	7030.40	7030.40	0.00	0.00	7604.00	7604.00	0.00	0.00
	20	13220.20	10133.32	-1.82	-10.19	13079.00	10152.71	0.12	-7.07	12344.40	9415.65	0.12	-6.00
	50	28486.60	14687.34	-1.41	-10.76	25671.80	14460.66	-1.31	-10.78	25430.20	14498.82	-0.16	-8.53
	80	43333.20	18614.36	-0.68	-20.95	37216.40	16575.84	0.41	-19.74	35280.60	16914.06	0.51	-15.04
	100	41128.80	17325.96	-0.79	-18.69	45392.80	19935.10	-1.60	-20.61	38959.20	17704.76	-0.60	-11.24
Average		26752.84	13671.28	-0.91	-12.12	25678.08	13630.94	0.06	-11.64	23923.68	13227.46	0.06	-8.16
Medium	10	7923.20	7923.20	0.00	0.00	7252.60	7252.60	0.00	0.00	7788.40	7480.43	0.00	-3.43
	20	13740.60	9898.95	-0.42	-12.59	13308.20	10108.08	-0.72	-5.70	12867.60	9337.96	0.51	-8.94
	50	30696.00	14389.70	0.36	-12.69	26440.80	14308.14	0.99	-12.30	25998.40	14284.88	0.38	-10.15
	80	45270.60	18403.18	0.08	-25.55	39469.60	16538.08	-0.15	-22.19	35673.40	16717.82	-0.15	-17.57
	100	40137.80	16331.96	-0.24	-25.86	39893.40	15819.10	0.28	-24.01†	34196.20	13671.20	-0.04	-16.27†
Average		27553.64	13389.40	-0.21	-15.34	25272.92	12805.20	-0.36	-12.84	23304.80	12298.46	0.25	-11.27
Large	10	7803.00	7803.00	0.00	0.00	7153.00	7153.00	0.00	0.00	7355.40	7355.28	0.00	0.00
	20	13795.20	9691.86	0.54	-14.79	13639.20	10022.46	0.12	-6.00	12881.00	9152.20	0.67	-8.34
	50	28859.00	14296.64	0.19	-12.87	26691.20	14223.02	-0.16	-8.53	26719.40	14201.98	-0.11	-10.64
	80	44374.80	18205.78	-0.21	-25.92	39197.00	16502.68	0.51	-15.04	36715.60	16832.36	-1.08	-15.17
	100	41836.60	17236.96	-0.38	-19.82	45939.60	19593.54	-0.60	-11.24	37997.80	16470.36	-0.07	-14.26
Average		27333.72	13446.85	0.27	-14.68	26524.00	13498.94	0.06	-8.16	24333.84	12802.43	0.33	-9.68
Global average		27213.40	13502.51	-0.28	-14.04	25825.00	13311.69	-0.08	-10.88	23854.11	12776.12	0.22	-9.70

†One instance (out of five) did not obtain a lower bound.

Compared to the results shown in Table 7 proving an initial solution guarantees obtaining an upper bound for all instances, even the large ones. Since this time no valid inequalities are present, the model contains fewer constraints and is, thus, to some extent lighter to optimize, leading to mostly equivalent solution quality (marginally better solutions for 1 depot instances with an improvement of 0.30%, improvement of 0.08% for 3-depot instances, and worse solution of about 0.22% for

5-depot instances).

Furthermore, as expected, providing valid inequalities improves the lower bounds considerably as indicated on the LB gap (%) of Table 8. In our case, compared to the average lower bounds presented in Tables 4–6, once the valid inequalities are removed from the model, the obtained LBs worsens dramatically (much lower values as presented in Table 8). Though ubiquitous, this effect are more evident for the larger instances.

These results help highlight the importance of both aspect developed in this paper: the valid inequalities to improve lower bounds, and the constructive heuristics to help provide feasible solutions even for very large instances of the problem.

4.2.3. Results of the proposed matheurisitc

The following three tables show the average results obtained by the proposed matheuristic, consisting of the constructive heuristic procedure followed by the set covering problem in which all non-dominated routes are used as potential route variables. We present a table per the number of potential depots on the instance and on the first two columns of each table, we provide general information on the instance. The information on the constructive heuristic is already presented in Section 3, however for the sake of simplicity of the analysis, they are presented again here on the third to fifth columns. From the fifth column on, we show the the improved results by applying the set covering. Therefore, we continue by reporting the number of non-dominated routes of the set covering phase, the solution (UB) obtained and its execution time in seconds (Time). Finally, the last two columns report the improvement over the heuristic and with respect to results obtained on Tables 4–6, respectively.

Several interesting observations can be drawn from the analysis of the results from Tables 9–11. First, we observe that the constructive heuristics exploit the sequencing of customers very well: from a huge number of routes generated, many are permutations of the same customers, leading to significantly fewer routes that are not dominated. This treatment has the advantage of reducing the computational burden when solving the set covering model. Moreover, these tables show that applying the set covering further improves the routes generated by the constructive heuristic. The improvement over the constructive heuristic solution is of 4.11% for instances with 1 depot, 4.48% for instances with 3 depots and 5.58% for instances with 5 depots. Moreover, the average execution time for the matheuristic is less that the time reported in Tables 4–6, which is insiginificant compared to the impact these solutions would have in practice.

Table 9: Average results on instances with 1 depot from the propsoed matheuristic

Instance	$\lvert N_c \rvert$	Constructive heuristic: # routes generated	Best solution	Time (s)	Set covering problem: # non-dominated routes	Solution	Time (s)	Improv. over heuristic (%)	Improv. over Section 4.2.2 (%)
Small	10	369078.40	7846.40	15	978.60	7770.40	<1	0.89	-2.53
	20	738245.00	14337.40	38	54055.60	13533.80	1	5.65	-0.52
	50	2479664.80	29915.40	220	591869.20	26917.40	23	10.06	6.60
	80	4182447.20	44227.80	1160	1164648.80	38568.40	326	12.77	11.49
	100	5308720.60	42219.60	1638	1606591.00	40974.20	244	2.72	0.95
	Average	2615631.20	27709.32	614	683628.64	25552.84	119	6.42	3.20
Medium	10	330806.20	8123.20	14	963.80	8071.80	<1	0.68	-1.69
	20	608242.60	14811.60	37	53929.40	13797.40	2	6.76	-0.18
	50	2165547.00	31184.20	204	696983.40	27134.00	38	12.84	11.17
	80	3793161.00	45658.20	707	1297387.40	39166.60	421	14.29	13.50
	100	4452738.00	40396.00	1132	1695888.40	39548.60	474	2.18	1.89
	Average	2270098.96	28034.64	419	749030.48	25543.68	187	7.35	4.94
Large	10	332077.20	7920.20	14	793.00	7859.60	<1	0.64	-0.81
	20	447788.40	14347.00	32	60496.60	13942.00	2	2.82	-1.54
	50	1803903.60	29372.20	190	650223.60	26158.80	35	10.76	8.98
	80	3318467.40	44811.40	530	1255734.80	39031.80	140	13.05	12.35
	100	4379811.00	42183.20	1115	1675409.00	41128.60	313	2.45	2.01
	Average	2056409.52	27726.80	376	728531.40	25624.16	98	5.94	4.19
Global average		2314046.56	27823.59	470	720396.84	25573.56	135	6.57	4.11

Table 10: Average results on instances with 3 depots from the propsoed matheuristic

Instance	$\lvert N_c \rvert$	Constructive heuristic: # routes generated	Best solution	Time (s)	Set covering problem: # non-dominated routes	Solution	Time (s)	Improv. over heuristic (%)	Improv. over Section 4.2.2 (%)
Small	10	1226114.40	7439.00	48	2820.40	7366.80	<1	0.94	-4.75
	20	2245948.20	13824.00	115	155854.60	13145.40	15	4.92	-0.70
	50	5918252.20	26654.20	591	1338632.80	23571.80	473	11.69	9.37
	80	10059494.80	37466.00	3263	2497688.40	33400.60	4959	10.97	9.98
	100	13027628.40	46539.40	4389	3246115.20	45278.00	4908	2.69	1.86
	Average	6495487.60	26384.52	1681	1448222.28	24552.52	2071	6.24	3.15
Medium	10	1009597.40	7512.60	39	2573.80	7443.80	<1	0.91	-2.66
	20	1401006.20	13785.00	92	187499.00	13317.60	18	3.37	0.63
	50	4990314.40	26914.80	531	1531736.20	23279.20	783	13.58	11.16
	80	9273097.20	39801.40	2978	2658364.00	34800.80	5996	12.62	12.04
	100	11595049.80	39929.80	4596	3890206.40	39407.80	5804	1.40	1.03
	Average	5653813.00	25588.72	1647	1654075.88	23649.84	2520	6.38	4.44
Large	10	995294.60	7467.40	40	2385.60	7419.20	<1	0.63	-3.72
	20	1361307.80	13887.80	89	183235.00	13376.60	14	3.67	1.90
	50	4763696.80	27027.20	522	1592500.20	23163.40	482	14.28	13.06
	80	8042322.80	39308.60	1397	2811714.00	33201.60	2991	15.65	15.04
	100	10303364.00	46000.00	4079	3876719.20	44527.20	6040	3.09	2.98
	Average	5093197.20	26738.20	1225	1693310.80	24337.60	1905	7.46	5.85
Global average		5747499.27	26237.15	1518	1598536.32	24179.99	2165	6.69	4.48

Table 9 summarizes the results for instances with 1 depot. As shown in this table, applying the set covering phase improves the average solution for all set of instances. The improvement is higher for the sets with more than 50 customers, for which combining different routes from different solutions can lead to superior solutions, improving the results obtained with the mathematical model by more than 13% in some cases. Sometimes, this phase is even capable of obtaining the optimal solution

Table 11: Average results on instances with 5 depots from the propsoed matheuristic

Instance	$\|N_c\|$	Constructive heuristic: # routes generated	Constructive heuristic: Best solution	Constructive heuristic: Time (s)	Set covering problem: # non-dominated routes	Set covering problem: Solution	Set covering problem: Time (s)	Set covering problem: Improv. over heuristic (%)	Set covering problem: Improv. over Section 4.2.2 (%)
Small	10	2037187.20	7904.20	68	4610.40	7842.80	1	0.91	-2.87
	20	3806241.80	12773.60	196	260662.00	12062.60	62	5.49	1.95
	50	9894822.60	26377.00	970	1970811.60	23196.40	993	12.05	8.97
	80	15631462.40	35763.80	5170	3667000.80	30949.00	4940	13.50	11.87
	100	20383988.40	39356.60	7734	5086044.80	38481.80	3114	2.04	1.67
	Average	10350740.48	24435.04	2828	2197825.92	22506.52	1822	6.80	4.32
Medium	10	1671495.00	7848.40	60	4418.00	7848.40	1	0.00	-1.05
	20	2625247.00	13113.40	160	287875.60	12594.20	72	3.96	1.59
	50	8029309.20	26475.00	862	2367104	23090.20	1287	12.77	10.76
	80	14230304.80	35950.20	5115	4395731	30520.20	5862	15.12	14.58
	100	17992807.60	34278.80	5828	6107402.40	33116.20	5011	2.94	2.78
	Average	8909832.72	23533.16	2405	2632506.24	21433.84	2447	6.96	5.73
Large	10	1647919.20	7479.80	58	4420.00	7479.80	1	0.00	-1.66
	20	2379870.00	12972.00	152	290219.60	12363.80	76	4.65	3.32
	50	8121700.00	27053.00	857	2291151.00	23300.40	1214	13.91	12.91
	80	12877642.00	37208.20	2281	4215957.40	31200.00	3969	16.14	15.87
	100	16530344.80	38183.20	5124	5856689.20	36843.00	5785	3.39	2.92
	Average	8311495.20	24579.24	1694	2531687.44	22237.40	2209	7.62	6.67
Global average		9190689.47	24182.48	2309	2454066.53	22059.25	2159	7.12	5.58

proved in Section 4.2.2. Similarly, Table 10 also shows how applying the set covering phase can improve the solution found by the constructive heuristic for all instances, and that the best average results, however, are obtained for instances with more than 50 customers. For the instances with five depots, as presented in Table 11, the optimization phase obtains the largest global average improvement (7.12%), and individual instances see an average improvement of over 15%, stressing the importance of combining different solutions in a smart way.

5. Conclusions

This paper investigates a very challenging and practical problem and it contributes to the integrated logistics literature as it presents the first mathematical formulation for the time-dependent location routing problem. To achieve quality solution in an acceptable computational time and to provide a pool of initial solutions, we have also proposed a matheuristic algorithm. We have compared the performance of our proposed algorithm against the exact method on instances generated with real data. We have also shown that to solve complex problems like this, it is important to integrate heuristic methods and exact formulations. Our constructive heuristic is able to generate diversified routes and initial solutions that can be combined and optimized through a set covering problem resolution. Moreover, our exact mathematical formulation is able to solve instances with up to 100 customers and 15 time intervals, where traffic condition changes hourly. The inclusion of a set of valid inequalities provides better lower bounds while a pool of initial solutions enables further improvements.

Acknowledgments

This project was partly funded by the Canadian Natural Science and Engineering Council (NSERC) under grant 2014-05764. This support is greatly acknowledged. We thank Calcul Québec for providing high performance parallel computing facilities, and Hamza Heni and Khaled Belhassine for their help on creating the instances.

References

D.E. Akyol and R.B.M. De Koster. Determining time windows in urban freight transport: A city cooperative approach. *Transportation Research Part E: Logistics and Transportation Review*, 118:34–50, 2018.

P. Alvarez, I. Lerga, A. Serrano-Hernandez, and J. Faulin. The impact of traffic congestion when optimising delivery routes in real time. A case study in Spain. *International Journal of Logistics Research and Applications*, 21(5):529–541, 2018.

M. Behnke and T. Kirschstein. The impact of path selection on GHG emissions in city logistics. *Transportation Research Part E: Logistics and Transportation Review*, 106:320–336, 2017.

T. Bektaş, T.G. Crainic, and T. Van Woensel. From managing urban freight to smart city logistics networks. In K. Gakis and P. Pardalos, editors, *Network Design and Optimization for Smart Cities*, chapter 7, pages 143–188. World Scientific, 2017.

K. Belhassine, L.C. Coelho, J. Renaud, and J.-P. Gagliardi. Improved home deliveries in congested areas using geospatial technology. Technical report, Working Paper CIRRELT-2018-02, January 2018.

E. Demir, T. Bektaş, and G. Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014.

M. Drexl and M. Schneider. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2):283–308, 2014.

M. Drexl and M. Schneider. A survey of the standard location-routing problem. *Annals of Operations Research*, 259(1-2):389–414, 2017.

S. Erdoğan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012.

M. Gendreau, G. Ghiani, and E. Guerriero. Time-dependent routing problems: a review. *Computers & Operations Research*, 64:189–197, 2015.

A.W.A. Hammad, A. Akbarnezhad, and D. Rey. Sustainable urban facility location: Minimising noise pollution and network congestion. *Transportation Research Part E: Logistics and Transportation Review*, 107:38–59, 2017.

L. Hu, J.X. Zhu, Y. Wang, and L.H. Lee. Joint design of fleet size, hub locations, and hub capacities for third-party logistics networks with road congestion constraints. *Transportation Research Part E: Logistics and Transportation Review*, 118:568–588, 2018.

S. Ichoua, M. Gendreau, and J.-Y. Potvin. Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research*, 144(2):379–396, 2003.

İ. Kara, G. Laporte, and T. Bektaş. A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3), 2004.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The impact of depot location, fleet composition and routing on emissions in city logistics. *Transportation Research Part B: Methodological*, 84:81–102, 2016.

R.V. Kulkarni and P.R. Bhave. Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1):58–67, 1985.

V. Leggieri and M. Haouari. A practical solution approach for the green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 104:97–112, 2017.

C. Malandraki and M.S. Daskin. Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science*, 26(3):185–200, 1992.

S. Mancini. A combined multistart random constructive heuristic and set partitioning based for mulation for the vehicle routing problem with time dependent travel times. *Computers & Operations Research*, 88: 290–296, 2017.

F. Maranzana. On the location of supply points to minimize transport costs. *Operational Research Quarterly*, 15(3):261–270, 1964.

M. Marufuzzaman and S.D. Ekşioğlu. Managing congestion in supply chains via dynamic freight routing: An application in the biomass supply chain. *Transportation Research Part E: Logistics and Transportation Review*, 99:54–76, 2017.

G. Nagy and S. Salhi. Location-routing: issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

M. Veenstra and L.C. Coelho. The time-dependent shortest path and vehicle routing problem. Technical report, Working Paper CIRRELT-2017-57, August 2017.

E. Von Boventer. The relationship between transportation costs and location rent in transportation problems. *Journal of Regional Science*, 3(2):27–40, 1961.

Y. Yuan and J. Yu. Locating transit hubs in a multi-modal transportation network: A cluster-based optimization approach. *Transportation Research Part E: Logistics and Transportation Review*, 114:85–103, 2018.