CIRRELT-2021-16

A Benders Decomposition Algorithm for the Time-Dependent Vehicle Routing Problem

Pedro B. Castellucci
Maryam Darvish
Leandro C. Coelho

April 2021

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2021-003

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

A Benders Decomposition Algorithm for the Time-Dependent Vehicle Routing Problem

Pedro B. Castellucci[1], Leandro C. Coelho[2,*], Maryam Darvish[2]

1. Universidade Federal de Santa Catarina, Brazil
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, Université Laval, Québec, Canada

Abstract. This paper presents an algorithm based on Benders decomposition to solve the time-dependent vehicle routing problem (VRP). By solving the master problem, capacity cuts are created but the time-dependency is ignored. Once a feasible VRP solution is obtained, a subproblem for each route is defined, whose solutions can create either optimality or infeasibility cuts to enforce time-dependent solutions. Through computational experiments, we demonstrate how this decomposition method can be applied to the classical two-index and three-index VRP formulations. The results indicate that the decomposition applied to the first formulation yields large gaps and requires very long computing times whereas when the decomposition is applied to the larger formulation, the quality of the generated cuts increase significantly. The dual bound improves drastically, and the solutions to the problem are then proven to be optimal or very tight.

Keywords: Vehicle routing problem, Benders decomposition, time-dependent.

Acknowledgements. This project was partly funded by the Natural Sciences and Engineering Council of Canada (NSERC) under grants 2019-00094 and 2020-00401. This support is greatly acknowledged.

---

* Corresponding author: Leandro.Coelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2021

1 Introduction

Distribution of goods and services is vital to the prosperity of urban areas but it also requires efficient planning and management of all activities. Among these activities, scheduling and routing of vehicles to satisfy the demand of customers still remains a challenge [37]. The literature has proposed several models and solution algorithms to design minimum-cost vehicle routes [35]. However, many of the problems studied hold oversimplifying assumptions. For example, the classic Vehicle Routing Problem (VRP) does not include traffic information, assuming that travel time is constant throughout the day. Peak-hour congestion or unforeseeable events, such as accidents, increase the travel time in urban areas.

Different travel speeds on road segments due to congestion result in varying travel time. By analyzing historical traffic data on a road network, one can estimate the patterns of traffic flow, congestion timing, and location. However, integrating this information into the VRP framework to solve VRPs with varying travel times requires very sophisticated solution algorithms. Moreover, not all intricate algorithms developed for the classic VRP have proved promising solutions for its time-dependent counterpart, i.e., the time-dependent VRP (TDVRP) [25].

The goal of this paper is to propose an exact solution algorithm for the TDVRP. We apply our proposed solution algorithm to solve real-world delivery problems from the literature, based on the road network in Quebec city, Canada. Our instances are created using the historical traffic data on road networks that consider speed and travel time on each road segment. The main contribution of this paper is to provide an exact method to solve large size instances of the TDVRP. To this end, we first present two formulations for the problem and evaluate their performances. We then propose a logic-based Benders decomposition algorithm. Our proposed solution algorithm relies on a branch-and-cut procedure, largely based on Benders decomposition [3]. While in traditional Benders decomposition, the subproblems take the form of simple linear programming (LP) problems, in logic-based Benders decomposition (LBBD) [16, 15], the subproblems can take any general form. The cuts implied from them are not automatically given by the solution of the subproblems, and must rather be developed for the problem at hand.

The paper is organized as follows. In Section 2, we present a review of the literature. A formal problem description and two mathematical models are presented in Section 3. Section 4 describes our solution method, based on a logic-based branch and Benders cut algorithm. Section 5 presents the results of our computational experiments. Finally, conclusions are drawn in Section 6.

2 Literature review

The TDVRP has become a popular research topic in recent years and has gained increasing attention. In this section, first we provide a brief overview of the TDVRP literature (Section 2.1) and then we describe some successful applications of the LBBD method (Section 2.2).

2.1 Vehicle routing problem with time-dependent travel time

A comprehensive review of the TDVRP can be found in [12]. Ahn and Shin [1] considered the arrival time to each node to be a monotonic function of the arrival time, thus respecting the First In, First Out (FIFO) property. They proposed several feasibility checks for the TDVRP with time windows (TDVRPTW) and present three heuristics based on modifying the already existing ones, insertion, savings and arc exchange heuristic. Malandraki and Daskin [23] proposed the first mixed integer formulation for the TDVRP. They introduced a step-function distribution of the travel time. However, the authors do not consider the nonpassing or the FIFO principle.

Ichoua et al. [18] solved the TDVRP with soft time windows by a tabu search heuristic. To respect the FIFO principle, the step function was used to define the speed distribution rather than the travel time distribution. Fleischmann et al. [11] presented a framework to obtain the time-varying traffic information from the city of

Berlin. This was the first research paper which used real traffic data for the TDVRP. They also optimized the start time of each tour after solving the TDVRP. Later, Donati et al. [8] applied the Multi Ants Colony System (MACS) and a local search method to solve the TDVRP with hard time windows. They tested the heuristic on generated instances and also on a real road network from Padua, Italy. Van Woensel et al. [36] determined the travel speed and, consequently, the traffic congestion based on queuing theory. They also optimized the departure time of the vehicles from the depot. They showed that ignoring the time dependency of the travel time leads to unrealistic solutions. This study is similar to that of Ichoua et al. [18], as both considered multiple road types. Soler et al. [32] transformed the TDVRPTW into an asymmetric capacitated VRP and then solved small size instances of the problem to optimality using already existing solution algorithms. Kok et al. [19] investigated the impact of several congestion avoidance strategies on the performance of vehicle route plans. These strategies included selecting alternative routes, changing the customer visit sequences, and changing the vehicle-customer assignments. They developed a modified Dijkstra algorithm and a restricted dynamic programming heuristic to solve the problem. Figliozzi [10] proposed an algorithm to solve the TDVRP. The paper also proposed new test instances based on those of Solomon [33]. In most of the TDVRP papers, a heuristic approach is developed to solve the problem. The branch-and-price method proposed by Dabia et al. [5] for the TDVRP with time windows is the only paper, to the best of our knowledge, that uses an exact method. Their decomposition method considered a set partitioning problem as the master problem, solved by a column generation method, and a time-dependent shortest path with resource constraints as the pricing problem solved by a tailored labeling algorithm. The objective of the paper was to minimize the duration of all routes where different departure times from the depot were allowed. The authors modified the VRPTW instances of Solomon [33] and could solve 63% of the instances with 25 customers, 38% of the instances with 50 customers, and 15% of the instances with 100 customers.

Mancini [24] challenged several assumptions in previous TDVRP studies, such as the use of simplified step functions, discretizing the time horizon in small time intervals, and considering Euclidean distances. The author studied the main road network of Torino with time varying travel time. Huang et al. [17] investigated the integration of precomputed path selection, stochastic traffic conditions, and flexible departure time from the depot. They addressed path flexibility in the TDVRP, using instances generated from the road network of Beijing. They use a Route-Path approximation method to generate near-optimal solutions under stochastic traffic conditions.

2.2 Logic-based Benders decomposition

Benders decomposition [3] is one of the most widely applied and successful decomposition approaches. Its main idea relies on decomposing and solving smaller continuous linear problems. However, when linear programming (LP) subproblems cannot be obtained, then standard linear duality cannot be applied to develop Benders cuts either [28]. Therefore, to deal with such cases, other types of cuts such as those derived from the LBBD [16, 15] are required to be used [29].

The LBBD and its variants are successfully applied to a wide range of problems. Only in the context of distribution optimization, several recent studies have demonstrated its efficiency, e.g., heterogeneous VRP with time windows [9], heterogeneous fixed fleet VRP based on fuel and carbon emissions [21], selective dial-a-ride problem [30], home healthcare delivery [14], the inventory routing problem with perishable products and environmental costs [2], and distribution network design [27].

3 Problem description and mathematical model

In this section, we provide both the two- and three-index-based formulations of the TDVRP. Our two-index formulation extends that of Laporte and Nobert [20] for the VRP, and is slightly different from the one proposed in Malandraki and Daskin

[23], as they consider time windows and forbid subtours with the Miller-Tucker-Zemlin (MTZ)-like constraints [26]. Our three-index formulation extends that of Golden et al. [13]. The TDVRP is defined on a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$ where $\mathcal{V}$ is the set of all nodes and $\mathcal{A} = \mathcal{V} \times \mathcal{V}$ is the set of arcs. Let 0 be the node of the depot and $\mathcal{V}'$ be the set of n customer nodes.

Each customer $i \in \mathcal{V}'$ has a service time s_i and a demand d_i to be satisfied by K vehicles of capacity Q from the set $\mathcal{K}$. The usage of a vehicle incurs a fixed cost f per vehicle, and a variable cost w per unit of time. Routes take place during a day which is divided into several intervals with a duration of t time units. At each interval $m \in \mathcal{M}$, a known and constant travel time (cost) along arc (i, j) is considered c_{ij}^m. The planning horizon is of length T, which indicates the latest time by which all vehicles must be back to the depot. Moreover, as auxiliary parameters, we define $c_{ij}^{min} = \min_{m \in \mathcal{M}} c_{ij}^m$ as the minimum travel time along arc (i, j).

To model the return of each vehicle k to the depot, we add, to set $\mathcal{V}$, dummy nodes $\{n + 1, \ldots, n + K\}$, representing the arrival nodes of K vehicles back to the depot.

The objective of the problem is to minimize the total fixed and variable costs, while meeting all customer demands and respecting several problem specific constraints.

We first model the problem using two sets of variables. Continuous variables z_i, $i \in \mathcal{V}$, represent the departure time of a vehicle from node $i \in \mathcal{V}' \cup \{0\}$ or its arrival time at any node $i \in \{n + 1, \ldots, n + K\}$. Routing variables x_{ij}^m are equal to one if a vehicle leaves node i at interval m towards node j, and zero otherwise. Table 1 presents the notation used.

3.1 Two-index-based formulation

The problem can be formulated as the following mixed-integer linear program, which extends the well-known two-index vehicle routing problem model of Laporte and Nobert [20].

Table 1: Sets, parameters and variables.

$\mathcal{V}$	set of nodes
$\mathcal{V}'$	set of customers
$\mathcal{A}$	set of arcs
$\mathcal{M}$	set of intervals
$\mathcal{K}$	set of vehicles
K	number of vehicles
Q	vehicle capacity
d_i	demand of customer i
s_i	service time at customer i
c_{ij}^m	travel time between nodes i and j in interval m
t	duration of each interval
f	fixed cost associated to vehicles
w	variable cost for using a vehicle per unit of time
T	length of the planning horizon
z_i	departure time of any vehicle from node $i \in \mathcal{V}' \cup \{0\}$ and arrival time of any vehicle to node $i \in \{n+1, \ldots, n+K\}$
x_{ij}^m	equal to 1 if a vehicle leaves node i toward node j in time interval m

$$\text{minimize} \sum_{j \in \mathcal{V}'} \sum_{m \in \mathcal{M}} f x_{0j}^m + \sum_{k \in \mathcal{K}} w z_{n+k} \tag{1}$$

subject to:

$$\sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{V}' \cup \{0\}} x_{ij}^m = 1 \quad j \in \mathcal{V}' \tag{2}$$

$$\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{V} \setminus \{0\}} x_{ij}^m = 1 \quad i \in \mathcal{V}' \tag{3}$$

$$\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{V}'} x_{0j}^m \leq K \tag{4}$$

$$\sum_{(i,j) \in \mathcal{S} \times \mathcal{S}} \sum_{m \in \mathcal{M}} x_{ij}^m \leq |\mathcal{S}| - \sum_{i \in \mathcal{S}} \frac{d_i}{Q} \quad \mathcal{S} \subseteq \mathcal{V} \tag{5}$$

$$\sum_{i \in \mathcal{V}'} \sum_{m \in \mathcal{M}} x_{ij}^m \leq 1 \quad j \in \{n+1, \ldots, n+K\} \tag{6}$$

$$x_{0j}^m = 0 \quad m \in \mathcal{M}, j \in \mathcal{V} \setminus \{\mathcal{V}'\} \tag{7}$$

$$z_0 = 0 \tag{8}$$

$$z_j \geq z_i + c_{ij}^m + s_j - (1 - x_{ij}^m)(T + c_{ij}^m + s_j) \quad i \in \mathcal{V}' \cup \{0\}, j \in \mathcal{V} \setminus \{0\}, m \in \mathcal{M} \tag{9}$$

$$z_i - t(m-1) x_{ij}^m \geq 0 \quad i \in \mathcal{V}' \cup \{0\}, j \in \mathcal{V} \setminus \{0\}, m \in \mathcal{M} \tag{10}$$

$$z_i \leq t \cdot m + (1 - x_{ij}^m)(T - t \cdot m) \quad i \in \mathcal{V}' \cup \{0\}, j \in \mathcal{V}, m \in \mathcal{M} \tag{11}$$

$$s_i + \min_{j \in \mathcal{V}, j \neq i} c_{ij}^{min} \leq z_i \leq T - c_{i0}^{min} \quad i \in \mathcal{V}' \tag{12}$$

$$x_{ij}^m \in \{0, 1\} \quad (i,j) \in \mathcal{A}. \tag{13}$$

The objective function (1) minimizes the fixed cost of using the available vehicles and their per time unit usage costs. Constraints (2) and (3) are degree and flow balance constraints. The number of vehicles is limited by constraint (4). Constraints (5) ensure the capacity of the vehicles is respected and eliminates subtours. Any dummy node representing the arrival depot for each vehicle can be visited at most once, as imposed by constraints (6). Constraints (7) and (8) indicate that the used vehicles must leave the depot toward customers at the beginning of the planning horizon. Departure time consistency at nodes visited successively is guaranteed by constraints (9). Lower and upper bounds on departure times, linking the actual departure time with its corresponding interval m in routing variables are defined in constraints (10) and (11). Constraints (12) and (13) define the domain of the variables.

This model is sufficient to correctly formulate the problem at hand. It can be strengthened by the use of the following valid inequalities that reinforce some of its constraints and impose lower bounds on the use of some of the variables.

We prevent subtours of two and three customers to make the formulation tighter, using constraints (14) and (15).

$$\sum_{m \in \mathcal{M}} x_{ij}^m + x_{ji}^m \leq 1 \quad i \in \mathcal{V}', j \in \mathcal{V}' \tag{14}$$

$$\sum_{m \in \mathcal{M}} x_{ij}^m + x_{ji}^m + x_{jp}^m + x_{pj}^m + x_{pi}^m + x_{ip}^m \leq 2 \quad i \in \mathcal{V}', j \in \mathcal{V}', p \in \mathcal{V}'. \tag{15}$$

As shown in [31], the minimum number of vehicles used can be computed by solving a bin packing problem as follows. Let $BP(d, Q)$ be an optimal solution of a bin packing problem with items of size $d = d_0, d_1, \ldots, d_n$ and bins with capacity Q. Then, we can add constraints (16) to the model. This helps increase the first term of the objective function (1).

$$\sum_{m \in \mathcal{M}} \sum_{j \in \mathcal{V}'} x_{0j}^m \geq BP(d, Q). \tag{16}$$

Also from [31], one can eliminate some variables, given that the minimum time to reach a particular node i plus the service time s_i is incompatible with using arc (i, j) in time period $m \in \mathcal{M}$. This translates to constraints (17) and (18).

$$x_{ij}^m = 0 \quad i \in \mathcal{V}' : s_i + \min_{a \in \mathcal{V}' \cup \{0\}, a \neq i} c_{ai}^m > m \cdot t, \quad j \in \mathcal{V}', m \in \mathcal{M} \tag{17}$$

$$x_{ij}^m = 0 \quad i \in \mathcal{V}' : s_i + \min_{a \in \mathcal{V}'} c_{0a}^m > m \cdot t, \quad j \in \mathcal{V}', m \in \mathcal{M}. \tag{18}$$

A minimum spanning tree (MST) considering the set of customers ($\mathcal{V}'$) may be used to provide a lower bound on the hourly cost of using vehicles (second term of the objective function (1)). Let $MST(\mathcal{V}')$ be the cost of a minimum spanning tree of nodes $\mathcal{V}'$, defined over a support graph where all arc traversal costs are set to the minimum observed in any time interval m, c_{ij}^{min}. This constitutes a lower bound on the cost of visiting all customers. Moreover, we can add the $BP(d,Q)$ arcs for the cheapest connections between the depot and any customer, for up to the lower bound obtained by solving the bin packing as described before: $\overset{BP(d,Q)}{\min_{j \in \mathcal{V}'}} c_{0j}^{min}$. Thus, we can use inequality (19) as a lower bound on the time use of the vehicles.

$$\sum_{m \in \mathcal{M}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} (c_{ij}^m + s_i) x_{ij}^m \geq \overset{BP(d,Q)}{\min_{j \in \mathcal{V}'}} c_{0j}^{min} + \overset{BP(d,Q)}{\min_{i \in \mathcal{V}'}} c_{i0}^{min} + MST(\mathcal{V}') + \sum_{i \in \mathcal{V}} s_i. \tag{19}$$

Finally, the present model is oblivious to the fact that all vehicles are homogeneous and that the subset of customers visited by each vehicle can be interchanged, creating symmetrical solutions that hinder branch-and-bound-based algorithms. To this end, we can add symmetry breaking constraints (20) that impose an order on the vehicles.

$$z_{n+1} \geq z_{n+2} \geq \ldots \geq z_{n+K}. \tag{20}$$

3.2 Three-index-based formulation

An alternative model to the previous TDVRP is based on the three-index of Golden et al. [13], here presented with the stronger Dantzig-Fulkerson-Johnson subtour elimination constraints [6]. To this end, and with a little abuse of notation, we define variables x_{ij}^{km} equal to one if and only if arc (i,j) is traversed by vehicle k in period m. This model decomposes the solution graph per vehicles, unlike the aggregated graph of the previous model.

$$\text{minimize} \sum_{j\in\mathcal{V}'}\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}} f x_{0j}^{km} + \sum_{k\in\mathcal{K}} w z_{n+k} \tag{21}$$

subject to:

$$\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}'\cup\{0\}} x_{ij}^{km} = 1 \quad j\in\mathcal{V}' \tag{22}$$

$$\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{V}\backslash\{0\}} x_{ij}^{km} = 1 \quad i\in\mathcal{V}' \tag{23}$$

$$\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{V}'\cup\{0\}} x_{ij}^{km} = \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{V}\backslash\{0\}} x_{ji}^{km} \quad j\in\mathcal{V}', k\in\mathcal{K} \tag{24}$$

$$\sum_{(i,j)\in\mathcal{S}\times\mathcal{S}}\sum_{m\in\mathcal{M}} x_{ij}^{km} \le |\mathcal{S}| - \sum_{i\in\mathcal{S}}\frac{d_i}{Q} \quad \mathcal{S}\subseteq\mathcal{V}, k\in\mathcal{K} \tag{25}$$

$$x_{0j}^{km} = 0 \quad m\in\mathcal{M}, j\in\mathcal{V}\backslash\{\mathcal{V}'\} \tag{26}$$

$$z_0 = 0 \tag{27}$$

$$\sum_{j\in\mathcal{V}'}\sum_{m\in\mathcal{M}} x_{0j}^{km} \le 1 \quad k\in\mathcal{K} \tag{28}$$

$$z_j \ge z_i + c_{ij}^m + s_j - (1 - x_{ij}^{km})(T + c_{ij}^m + s_j) \quad i\in\mathcal{V}'\cup\{0\}, j\in\mathcal{V}\backslash\{0\}, k\in\mathcal{K}, m\in\mathcal{M} \tag{29}$$

$$z_i - t(m-1)x_{ij}^{km} \ge 0 \quad i\in\mathcal{V}'\cup\{0\}, j\in\mathcal{V}\backslash\{0\}, k\in\mathcal{K}, m\in\mathcal{M} \tag{30}$$

$$z_i \le t\cdot m + (1 - x_{ij}^{km})(T - t\cdot m) \quad i\in\mathcal{V}'\cup\{0\}, j\in\mathcal{V}, k\in\mathcal{K}, m\in\mathcal{M} \tag{31}$$

$$s_i + \min_{j\in\mathcal{V}, j\neq i} c_{ij}^{min} \le z_i \le T - c_{i0}^{min} \quad i\in\mathcal{V}' \tag{32}$$

$$x_{ij}^m \in \{0,1\} \quad (i,j)\in\mathcal{A}. \tag{33}$$

For conciseness, we do not describe this model at length as it is very similar to the previous one. For the same reason, we do not reproduce here the extended forms of inequalities (14)–(20) that can also be applied to this model. The following valid inequalities are also added to this model.

Constraints (34) associate each vehicle with a particular dummy node. Constraints (35) define lower bounds for travel costs using the fact that a vehicle must, at least, visit a customer and return to the depot. Inequalities (36) forbid routes that exceed the length of the planning horizon.

$$\sum_{j\in\mathcal{V}'}\sum_{m\in\mathcal{M}} x_{0j}^{km} = \sum_{i\in\mathcal{V}'}\sum_{m\in\mathcal{M}} x_{i,n+k}^{km} \quad k\in\mathcal{K} \tag{34}$$

$$\sum_{i\in\mathcal{V}'\cup\{0\}}\sum_{j\in\mathcal{V}\setminus\{0\}}\sum_{m\in\mathcal{M}} c_{ij}^{min} x_{ij}^{km} \geq \min_{j\in\mathcal{V}':j>k} c_{0j}^{min} + \min_{j\in\mathcal{V}':i>k} c_{i,n+k}^{min} \quad k\in\{1,\ldots,BP(d,Q)\} \tag{35}$$

$$\sum_{i\in\mathcal{V}'\cup\{0\}}\sum_{j\in\mathcal{V}\setminus\{0\}}\sum_{m\in\mathcal{M}} (c_{ij}^{m}+s_i) x_{ij}^{km} \leq T \quad k\in\mathcal{K}. \tag{36}$$

Moreover, some symmetry breaking constraints can be added to this formulation, inspired from Darvish et al. [7]. Considering the set $\mathcal{K}$ of vehicles, there is an optimal solution that uses the first vehicle for the first route, the second vehicle for the second one, and so on. That is, vehicles have a priority to be used. This yields constraints (37).

$$\sum_{j\in\mathcal{V}'} x_{0j}^{k0} \leq \sum_{j\in\mathcal{V}'} x_{0j}^{k-1,0} \quad k\in\mathcal{K}\setminus\{1\}. \tag{37}$$

Also, if a customer j is visited by vehicle $k>1$, then, we can impose that at least one customer $p<j$ is visited by vehicle $k-1$. This translates to constraints (38).

$$\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{V}'\cup\{0\}} x_{ij}^{km} \leq \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{V}'\cup\{0\}}\sum_{p\in\mathcal{V}':p<j} x_{ip}^{k-1,m} \quad j\in\mathcal{V}', k\in\mathcal{K}\setminus\{1\}. \tag{38}$$

Finally, if vehicle $k>1$ visits customer j, then, at least $k-1$ vehicles are used to visit customers $1,2,\ldots,j-1$, which can be expressed by constraints (39).

$$\sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{V}'\cup\{0\}} k x_{ij}^{km} \leq \sum_{m\in\mathcal{M}}\sum_{i\in\mathcal{V}'\cup\{0\}}\sum_{q\in\mathcal{V}':q<j}\sum_{\bar{k}\in\mathcal{K}:\bar{k}<k} x_{iq}^{\bar{k}m} \quad j\in\mathcal{V}', k\in\mathcal{K}\setminus\{0\}. \tag{39}$$

Note that constraints (39) imply that $x_{ij}^{km} = 0$, $i \in \mathcal{V}' \cup \{0\}$, $j \in \mathcal{V}'$, $k \in \mathcal{K}$, $k > j$.

4 Solution methods

Both formulations presented in Section 3 can be solved by a branch-and-cut algorithm when constraints (5) and (25) are added dynamically. In order to apply the LBBD algorithm, the problem must be decomposed into a master problem and the subproblem(s). In our case, our master problem determines routes by selecting which arcs to travel, but ignores the information regarding traffic, i.e., the master problem is a relaxation of the TDVRP in which arc traversal costs are constant throughout the day. Our subproblems, one per vehicle, compute the actual arrival and departure times at each node considering the time-dependent cost matrix.

Specifically, whenever the master problem obtains an integer feasible solution, i.e., a solution without subtours, subproblems are invoked to compute the actual timing variables. For each subproblem, two situations may arise: *(i)* the subproblem is feasible and optimality cuts are generated to indicate to the master problem the optimal cost of that sequence of customers, or *(ii)* the subproblem is infeasible due to time horizon violation, in this case a feasibility cut is generated to indicate to the master problem that the subset of customers of that route cannot be visited following that particular sequence.

Moreover, unlike traditional Benders decomposition in which the master problem has to be solved from scratch at each iteration, here optimality and feasibility cuts are added in a branch-and-cut fashion, which is known as branch-and-check or branch and Benders cut [15, 4, 34]. To this end, the master problem is solved only once, and during its resolution, Benders cuts are added to the nodes of its branching tree, similar to a traditional branch-and-cut algorithm.

More formally, we first reformulate our model to make it amendable to this procedure. To this end, in Section 4.1 we present the master problem obtained for each formulation, and in Section 4.2 we define the subproblems of our algorithm.

4.1 Benders master problem

The problem can be reformulated by enumerating all possible routes $\mathcal{R}_k$ for vehicle k. A route r is defined as a sequence of nodes $< 0, i_1, i_2, \ldots, 0' >$ departing from the depot node 0, visiting some customer nodes, and ending at the depot, here indicated as $0'$; this node is in fact one of the $n+1, \ldots, n+K$ nodes associated with a vehicle, and the route (and its cost) remains the same regardless of which vehicle performs it. For any route $r \in \mathcal{R}$, let $\overline{z}^r_{n+k}$ be the additional time-dependent travel time of vehicle k performing this route. Let $\mathcal{A}_r$ be the set of arcs used in route $r \in \mathcal{R}$. Initially, set $\mathcal{R}$ is empty. Moreover, let $\mathcal{R}^{inf}$ be the set of routes that are infeasible due to time limit constraints, also initially empty. We define variables $x_{ij} \in \{0,1\}$ to indicate whether arc $(i,j) \in \mathcal{A}$ is traversed and $\eta_k \geq 0$ as the total cost of using vehicle $k \in \mathcal{K}$. The master problem reformulation (40)–(46) based on the two-index VRP model is given by:

$$\text{minimize} \quad \left(\sum_{j \in \mathcal{V}'} f x_{0j} + \sum_{i \in \mathcal{V}' \cup \{0\}} \sum_{j \in \mathcal{V} \setminus \{0\}} w c^{min}_{ij} x_{ij} + \sum_{k \in \mathcal{K}} \eta_k \right) \tag{40}$$

subject to:

$$\sum_{i \in \mathcal{V}' \cup \{0\}} x_{ij} = 1 \quad j \in \mathcal{V}' \tag{41}$$

$$\sum_{j \in \mathcal{V} \setminus \{0\}} x_{ij} = 1 \quad i \in \mathcal{V}' \tag{42}$$

$$\sum_{j \in \mathcal{V}'} x_{0j} \leq K \tag{43}$$

$$\sum_{(i,j) \in \mathcal{S} \times \mathcal{S}} x_{ij} \leq |\mathcal{S}| - \sum_{i \in \mathcal{S}} \frac{d_i}{Q} \quad \mathcal{S} \subseteq \mathcal{V} \tag{44}$$

$$x_{0j} = 0 \quad j \in \mathcal{V} \setminus \{\mathcal{V}'\} \tag{45}$$

$$x_{ij} \in \{0,1\} \quad (i,j) \in \mathcal{A} \tag{46}$$

$$\eta_k \geq 0 \quad k \in \mathcal{K}. \tag{47}$$

The objective function (40) and constraints (41)–(44) are equivalent to (1)–(5). Constraints (45) forbid a vehicle to leave the depot directly back to the depot, and constraints (46) and (47) define the nature of the decision variables. We also reinforce this model with constraints (14)–(19).

Model (40)–(47) is essentially a model for the VRP defined with arc costs c_{ij}^{min}. To properly account for the additional (time-dependent) cost of traversing each arc, for each feasible solution of (40)–(47), we generate optimality cuts (48).

$$\eta_k \geq \overline{z}_{n+k}^r - \overline{z}_{n+k}^r \sum_{(i,j)\in\mathcal{A}_r} (1 - x_{ij}) \quad k \in \mathcal{K}, r \in \mathcal{R} \tag{48}$$

Also, a feasible solution for (40)–(47), might be infeasible due to time limitation constraints. In this case, we generate feasibility cuts (49) or (50).

$$\sum_{i\in\mathcal{V}'\cup\{0\}} \sum_{j\in\mathcal{V}\setminus\{0\}} (c_{ij}^{min} + s_i)x_{ij} \leq T \quad (i,j) \in \mathcal{A}_r, r \in \mathcal{R}, \tag{49}$$

$$\sum_{(i,j)\in\mathcal{A}_r} x_{ij} \leq |\mathcal{A}_r| - 1 \quad (i,j) \in \mathcal{A}_r, r \in \mathcal{R}^{inf}. \tag{50}$$

Note that constraints (49) only has the information regarding minimum travel time, whilst (48) and (50) require time-dependent information. The construction of these cuts is explained in Section 4.2.

Model (40)–(47) has an exponential number of constraints due to (44) and (48)–(50). To handle capacity and subtour elimination constraints, (44) we can use a branch-and-cut algorithm where these constraints are dynamically generated whenever they are found to be violated. This can be done by using well-known methods such as those described in Lysgaard et al. [22].

A master problem derived from the three-index model is also possible, as presented next.

$$\text{minimize} \sum_{k\in\mathcal{K}} \left(\sum_{j\in\mathcal{V}'} f x_{0j}^k + \sum_{i\in\mathcal{V}'\cup\{0\}} \sum_{j\in\mathcal{V}'\cup\{0\}} w c_{ij}^{min} x_{ij}^k + \eta_k \right) \tag{51}$$

subject to:

$$\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}'\cup\{0\}} x_{ij}^k = 1 \quad j\in\mathcal{V}' \tag{52}$$

$$\sum_{k\in\mathcal{K}}\sum_{j\in V\cup\{0\}} x_{ij}^k = 1 \quad i\in\mathcal{V}' \tag{53}$$

$$\sum_{i\in\mathcal{V}} x_{ij}^k = \sum_{i\in\mathcal{V}} x_{ji}^k \quad j\in\mathcal{V}'\cup\{0\}, k\in\mathcal{K} \tag{54}$$

$$\sum_{j\in\mathcal{V}'} x_{0j}^k \le 1 \quad k\in\mathcal{K} \tag{55}$$

$$\sum_{(i,j)\in\mathcal{S}\times\mathcal{S}} x_{ij}^k \le |\mathcal{S}| - \sum_{i\in\mathcal{S}}\frac{d_i}{Q} \quad \mathcal{S}\subseteq\mathcal{V}, k\in\mathcal{K} \tag{56}$$

$$x_{ij}^k \in\{0,1\} \quad (i,j)\in\mathcal{A}, k\in\mathcal{K}, \tag{57}$$

$$\eta_k \ge 0 \quad k\in\mathcal{K}. \tag{58}$$

The objective function (51) minimizes the number of vehicles used and the hourly cost of each vehicle. The hourly cost is separated into minimum cost (c_{ij}^{min}) and additional cost (η_k). The constraints for this model are not described at length for conciseness. The valid inequalities presented in Section 3 remain valid. Besides, to make the formulation tighter we included *a priori* constraints (59) and (60) which account for subtours of two and three nodes.

$$x_{ij}^k + x_{ji}^k \le 1 \quad i\in\mathcal{V}', j\in\mathcal{V}', k\in\mathcal{K} \tag{59}$$

$$x_{ij}^k + x_{ji}^k + x_{jp}^k + x_{pj}^k + x_{pi}^k + x_{ip}^k \le 2 \quad i\in\mathcal{V}', j\in\mathcal{V}', p\in\mathcal{V}', k\in\mathcal{K}. \tag{60}$$

The number of vehicles used can be computed using a bin packing problem, similarly to what was done before. Then, we can add constraints (61) to the model.

$$\sum_{i\in\mathcal{V}'} x_{0jk} \ge BP(d,Q). \tag{61}$$

We can also add lower bounds for the hourly cost of using a vehicle. In the best case, each vehicle k goes to an unvisited consumer that is the closest to the depot with distance $\min\limits_{j \in \mathcal{V}': j > k} \{c_{0j}^{min}\}$ plus the trip back $\min_{i \in \mathcal{V}': i > k} \{c_{i0}^{min}\}$. Thereby, we have constraints (62).

$$\sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}'} c_{ij}^{min} x_{ij}^k \geq \min_{j \in \mathcal{V}': j > k} \{c_{0j}^{min}\} + \min_{i \in \mathcal{V}': i > k} \{c_{i0}^{min}\} \quad k \in \{1, \dots, BP(d, Q)\}. \tag{62}$$

Finally, a minimum spanning tree considering the set of customers ($\mathcal{V}'$) may be used to obtain a lower bound on the hourly cost of using the vehicles. Let $MST(\mathcal{V}')$ be the cost of a minimum spanning tree of nodes $\mathcal{V}'$ and $\min_{j \in \mathcal{V}'}^{BP(d,Q)} c_{0j}^{min}$ the $BP(d, Q)$ smallest distances connecting the depot and one customer per vehicle. Then, we have constraint (63).

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} (c_{ij}^{min} + s_i) x_{ij}^k \geq \min_{j \in \mathcal{V}'}^{BP(d,Q)} \{c_{0j}^{min}\} + \min_{i \in \mathcal{V}'}^{BP(d,Q)} \{c_{iK}^{min}\} + MST(\mathcal{V}') + \sum_{i \in \mathcal{V}'} s_i. \tag{63}$$

Similar to the two-index master problem, model (51)–(58) is essentially a VRP defined with arc costs c_{ij}^{min}. To properly, account for time-dependent travel time, for each feasible solution of (51)–(58), we generate either optimality cuts (64) or feasibility cuts (65).

$$\eta_k \geq \overline{z}_k^r - \overline{z}_k^r \sum_{(i,j) \in \mathcal{A}_r} (1 - x_{ij}^k) \quad k \in \mathcal{K}, r \in \mathcal{R}_k \tag{64}$$

$$\sum_{(i,j) \in \mathcal{A}_r} x_{ij}^k \leq |\mathcal{A}_r| - 1 \quad (i, j) \in \mathcal{A}_r, k \in \mathcal{K}, r \in \mathcal{R}^{inf}. \tag{65}$$

Note that for this model, we can add cuts (36) *a priori* to the master problem.

4.2 Benders subproblems

For each feasible solution of the master problem, we have a set of used routes $\overline{\mathcal{R}}_k$, one route for vehicle k. For each route $r \in \overline{\mathcal{R}}_k$, let $\mathcal{V}_r^k$ be the set of nodes it visits.

The objective function is to minimize the return time of vehicle k to the depot. For each arc (i, j) used in the master problem, i.e., $x_{ij} = 1$, we ensure the same arc is used in the subproblem via constraints (67). The timing relation between two consecutive visits is given by (68), and the link between x and z variables is done by constraints (69) and (70). The vehicle leaves the depot at time 0 as indicated by constraints (71). The nature and domain of the variables are enforced by constraints (72) and (73).

$$(SUB_r^k) \text{ minimize } wz_{n+k} \tag{66}$$

subject to:

$$\sum_{m \in \mathcal{M}} x_{ij}^{km} = 1 \quad (i, j) \in \mathcal{A}_r^k \tag{67}$$

$$z_j \geq z_i + c_{ij}^m + s_j - (1 - x_{ij}^{km})(T + c_{ij}^m + s_j) \quad (i, j) \in \mathcal{A}_r^k, m \in \mathcal{M} \tag{68}$$

$$z_i \leq t \cdot m + (1 - x_{ij}^{km})(T - t \cdot m) \quad (i, j) \in \mathcal{A}_r^k, m \in \mathcal{M} \tag{69}$$

$$z_i - t(m-1)x_{ij}^{km} \geq 0 \quad (i, j) \in \mathcal{A}_r^k, m \in \mathcal{M} \tag{70}$$

$$z_0 = 0 \tag{71}$$

$$z_i \geq 0 \quad i \in \mathcal{V}_r \tag{72}$$

$$x_{ij}^{km} \in \{0, 1\} \quad (i, j) \in \mathcal{A}_r^k, m \in \mathcal{M}. \tag{73}$$

The optimal value of z_{n+k} defines the arrival time of the vehicle after visiting all nodes in $\mathcal{V}_r$ following the arcs in $\mathcal{A}_r^k$. The value of $\overline{z}_k$ used in constraints (48) and (64) is obtained as $\overline{z}_k = z_{n+k} - \sum_{(i,j) \in \mathcal{A}_r^k} c_{ij}^{min}$.

Figure 1 provides a graphic summary of the decomposition algorithm. It begins by solving the master problem, which generates capacity cuts (e.g., constraints (5)) in a lazy constraint fashion. Once we have a feasible solution, the subproblems are defined, one for each route. If a subproblem is feasible, we generate an optimality cut (e.g., constraints (48)), otherwise we generate a feasibility cut (constraints (50)).

Furthermore, one may add a lifting procedure (not shown in Figure 1) to the generation of feasibility cuts the form (50) or (65). This consists of removing the node which leads to most savings (in time) to the route: if the solution remains infeasible, we have a stronger cut and can repeat the process. When a feasible solution is found, we also generate the corresponding optimality cut.

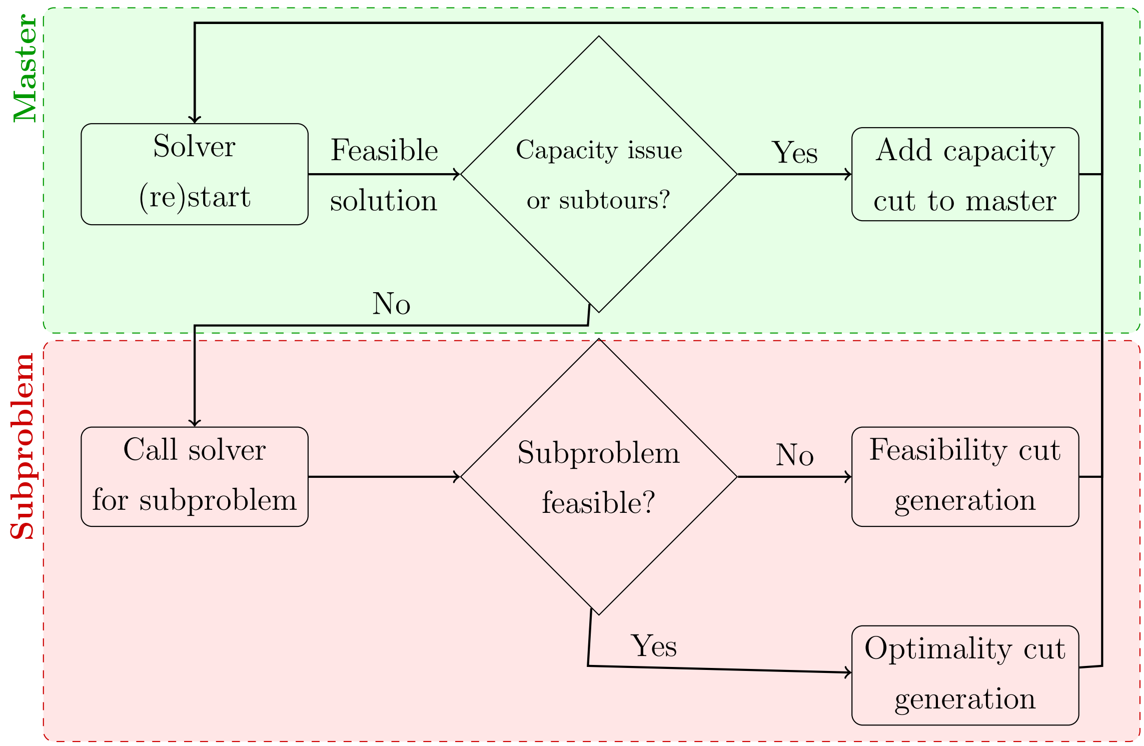


Figure 1: Schematic representation of the decomposition algorithm.

5 Computational experiments

The main goal of the computational experiments is to compare the performances of different proposed solution approaches: *(i)* two-index model with a branch-and–cut solver (Model-2), *(ii)* three-index model with a branch-and-cut solver (Model-3), *(iii)* the decomposition algorithm for the two-index model (Benders-2), *(iv)* the decomposition algorithm for the three-index model (Benders-3) and *(v)* Benders-3 with the refined feasibility cuts mentioned in Section 4.2 (Benders-3*).

For the branch-and-cut solver, we used Gurobi 9.0.2, with a time limit of 3600 seconds and eight threads. Other parameters were set to default value, including the MIP optimality gap (0.0001). For iteractively adding violated capacity constraints and the optimality and feasibility cuts of the decomposition algorithms we used the *lazy callback* feature. The algorithms were implemented in C++ and compiled with *g++* 7.5.0. Regarding hardware settings, we used an Intel Core i9-9900K with a frequency of 3.6 GHz with 16 processors and 128GB of RAM.

For our computational experiments, we use the the instances from [31] which are based on geographical information from real road traffic of Québec City (Canada). We used subsets with 10, 20 and 50 customers. In each subset, there are 30 instances grouped by the number of time intervals: *(i)* with five time intervals (*small*), *(ii)* with ten time intervals (*medium*), and *(iii)* with 15 time intervals (*large*). Customer demands were randomly generated from $[50, 750]$ units and service times (in seconds) from $[1000, 10800]$. Instances with 10 and 20 customers have a vehicle capacity of 4000 units and instances with 50 customers have a vehicle capacity of 4500 units. The fixed cost of using a vehicle is 1000 monetary units. The instances from [31] do not define a variable cost of vehicle usage, we define it as $w = 0.12$.

5.1 Comparison between solution approaches

Overall, Model-2 outperformed Model-3 as shown by a summary of the results in Table 2. Even though the time limit was reached for every instance when using any of the models, lower, upper and root bounds were consistently better (or the same) when using Model-2 for instances with 10 and 20 customer. This was observed in all of the instances for the lower and root bounds and in 28 (out of 30) for the upper bound. Also, the solver could find feasible solutions for all the instances with 10 customers for Model-2 and Model-3. However, Model-2 was more effective in finding feasible solutions for instances with 20 customers, finding them for 17 instances, against 7 when using Model-3. For instances with 50 customers, not a single feasible solution was found within the time limit with either Model-2 or Model-3.

Table 2: Summary of the results for Model-2 and Model-3. The table shows upper bound (UB), lower bound (LB), root bound (RB), gap ($100(UB - LB)/UB$), time to solve (in seconds), number of optimal solutions (#Opt) and number of feasible solutions (# Feas).

	Customers		UB	LB	RB	Gap	Time	# Opt	# Feas
Model-2	**10**	**Average**	9860.63	5871.05	2215.92	41.12	TL	0	30
		Median	9796.64	5745.57	2568.85	38.68	TL		
	20	**Average**	20616.18	3666.30	3091.30	82.10	TL	0	17
		Median	20419.70	3932.40	2809.96	84.42	TL		
	50	**Average**	Time limit reached without any feasible solution found for all of the instances						
		Median							
Model-3	**10**	**Average**	9965.73	3063.77	1659.09	69.49	TL	0	30
		Median	10010.40	2906.70	2000.00	71.19	TL		
	20	**Average**	21073.46	2574.50	2428.57	87.83	TL	0	7
		Median	20134.70	2000.00	2000.00	89.72	TL		
	50	**Average**	Time limit reached without any feasible solution found for all of the instances						
		Median							

Using the decomposition algorithm based on Model-2 (Benders-2) is competitive with Model-2 for instances with 20 customers (Table 3). For instances with 10 customers, the upper bounds of the two approaches do not deviate more than 1% of one another on average (and median). However, the lower bounds provided by Model-2 were higher for all instances, which lead to smaller optimality gaps, consistently. For instances with 20 customers, the decomposition (Benders-2) leads to better upper, lower and root bounds, resulting in smaller gaps on average (and median). Another advantage of the decomposition against Model-2 is the number of feasible solutions found for the case with 20 customers: 30 against 17 of Model-2. The number of feasibility cuts (49) was at least 30 times higher than the number of optimality cuts (50). The details on number of cuts added are shown later.

Benders-3 and Benders-3* decomposition algorithms consistently outperformed Model-2, Model-3 and Benders-2. Table 4 compares the average and median results

Table 3: Summary of the results for Model-2 and Benders-2. The table shows upper bound (UB), lower bound (LB), root bound (RB), gap ($100(UB - LB)/UB$), time to solve (in seconds), number of optimal solutions (# Opt) and number of feasible solutions (# Feas).

	Customers		UB	LB	RB	Gap	Time	# Opt	# Feas
Model-2	**10**	**Average**	9860.63	5871.05	2215.92	41.12	TL	0	30
		Median	9796.64	5745.57	2568.85	38.68	TL		
	20	**Average**	20616.18	3666.30	3091.30	82.10	TL	0	17
		Median	20419.70	3932.40	2809.96	84.42	TL		
	50	**Average**	Time limit reached without any feasible solution found for all of the instances						
		Median							
Benders-2	**10**	**Average**	9836.52	2548.87	2420.95	73.77	TL	0	30
		Median	9787.40	2857.16	2734.76	72.73	TL		
	20	**Average**	18680.40	3869.31	3822.23	79.20	TL	0	30
		Median	18805.56	3808.99	3780.34	79.66	TL		
	50	**Average**	Time limit reached without any feasible solution found for all of the instances						
		Median							

of Benders-2 and Benders-3. Regarding the upper bounds, the difference between the two approaches is smaller than 1% on average (and median). However, a significant difference in performance was observed in lower and root bounds. The improvement on these bounds allowed proof of optimality for all instances with 10 customers when using Benders-3 within 65 seconds on average (the maximum solution time was lower than 500 seconds). For instances with 20 customers, Benders-3 also reached the time limit for all instances, but the improvement on final gaps was more than ten times on average (and median). Furthermore, Benders-3 was the only approach that could find feasible solutions for instances with 50 customers. Similar analysis can be conducted focusing on Benders-3*. Even though Benders-3* was slightly slower on average (and median) than Benders-3, it showed a similar performance related to obtained bounds, feasible and optimal solutions. The experiments did not reveal a clear dominance of Benders-3 over Benders-3* or vice-versa.

The number of cuts generated by the Benders-2 and Benders-3 (or Benders-3*) is considerably different (Table 5). The number of cuts (49) can help explain the

Table 4: Summary of the results for Benders-2, Benders-3 and Benders-3*. The table shows upper bound (UB), lower bound (LB), root bound (RB), gap $(100(UB - LB)/UB)$, time to solve (in seconds), number of optimal solutions (# Opt) and number of feasible solutions (# Feas).

	Customers		**UB**	**LB**	**RB**	**Gap**	**Time**	**# Opt**	**# Feas**
Benders-2	**10**	**Average**	9836.52	2548.87	2420.95	73.77	TL	0	30
		Median	9787.40	2857.16	2734.76	72.73	TL		
	20	**Average**	18680.40	3869.31	3822.23	79.20	TL	0	30
		Median	18805.56	3808.99	3780.34	79.66	TL		
	50	**Average**	Time limit reached without any feasible solution found for all of the instances						
		Median							
Benders-3	**10**	**Average**	9831.63	9831.52	9280.65	0.00	64.62	30	30
		Median	9783.98	9783.98	9134.43	0.00	40.60		
	20	**Average**	18600.87	17725.00	17627.44	4.76	TL	0	30
		Median	18742.31	17902.15	17758.73	4.71	TL		
	50	**Average**	48137.89	43659.13	43614.85	9.27	TL	0	14
		Median	47830.90	43656.62	43586.03	9.07	TL		
Benders-3*	**10**	**Average**	9831.63	9831.50	9280.61	0.00	82.87	30	30
		Median	9783.98	9783.98	9134.43	0.00	46.50		
	20	**Average**	18604.72	17722.95	17627.50	4.79	TL	0	30
		Median	18754.37	17900.16	17758.73	4.75	TL		
	50	**Average**	48048.95	43565.18	43523.64	9.29	TL	0	14
		Median	47760.16	43631.59	43586.03	8.97	TL		

good performance of Benders-3, since they are integrated directly into the model in the form of (36) as mentioned in the end of Section 4.1. The number of feasibility cuts generated by Benders-3 and Benders-3* is considerably lower, which is to some extent expected by the addition of similar cuts *a priori* to Benders-3 (and Benders-3*) master problem. The difference in the number of optimality cuts however is hard to explain, but it can be related to the underlying structure of the resulting master problem or the effectiveness of the solver heuristic in providing good feasible solutions early in the solving process.

Table 5: Number of cuts added during the solution procedure for Benders-2, Benders-3 and Benders-3*. The table shows the number of optimality cuts, feasibility cuts and length cuts (only for Benders-2*).

	Customers		# (48) **or** (64)	# (50) **or** (65)	# (49)
Benders-2	**10**	**Average**	137615.00	1893.93	57019.50
		Median	160086.00	1433.50	59077.50
	20	**Average**	56367.87	655.53	32112.80
		Median	61086.00	613.50	19879.00
Benders-3	**10**	**Average**	5796.03	298.87	
		Median	2421.00	8.00	
	20	**Average**	13676.57	988.07	
		Median	11914.50	748.50	
	50	**Average**	2530.36	96.14	
		Median	2411.50	94.00	
Benders-3*	**10**	**Average**	6287.03	309.40	
		Median	2908.00	15.00	
	20	**Average**	14522.57	1007.63	
		Median	12487.50	885.00	
	50	**Average**	2749.07	103.21	
		Median	2499.00	89.00	

To further characterize the decomposition algorithms, we analyze the evolution of the bounds and the calls to the cut generation routines (optimality and feasibility).

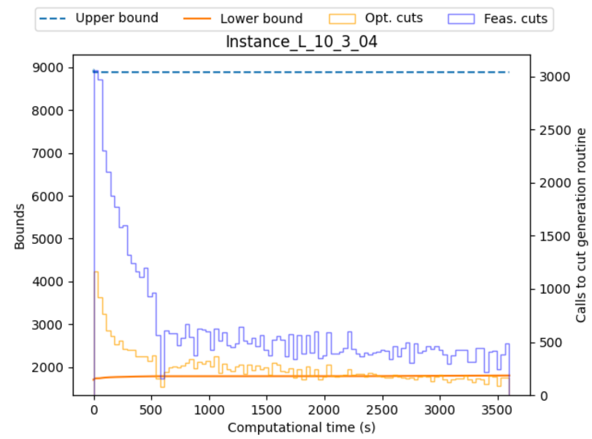
Figure 2 shows the typical behavior using two instances as examples for Benders-2 and Benders-3. In the beginning of the solution process, we usually have a high frequency of calls then this frequency drops (as shown in Figures 2a–2d). For instances in which the solver was able to prove optimality (e.g., Figure 2b), the frequency increases again near the end of the solution process. Typically, an improvement in the lower bound is associated with a large number of feasibility cuts (as evidenced by the first seconds of each execution and also clearly demonstrated when optimality is proven in Figure 2b. This corroborates the earlier suggestion that the incorporation of feasibility cuts (49) (in the form of (36)) into the Benders-3 master problem plays a crucial role in making its lower bound be so much tighter than any alternative.

The number of time intervals in an instance set (small, medium or large) was not a reliable predictor of performance (Figures (3) and (4)). For Model-2, we observed a higher median gap for small instances when compared to medium and large instances, but there are some overlaps between the distribution of values among other cases as shown in Figure 3. It also shows that for Model-3 and Benders-2, the number of time intervals considered in the instances does not reflect a difference in performance.

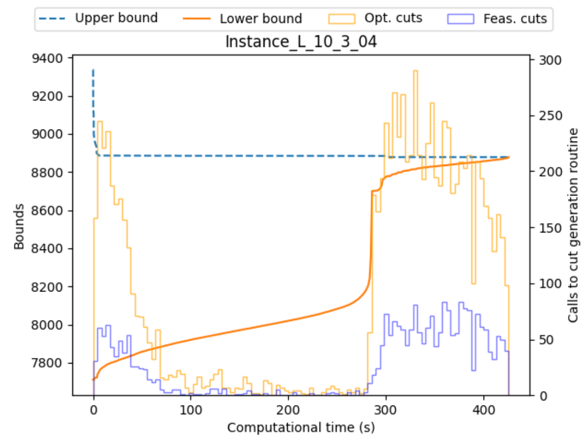
A similar effect was observed for Benders-3 approach (Figure 3). For instances with 20 customers, the distribution of values is similar when comparing small, medium and large instances. The same effect happened for instances with 50 customers, even though, in this case, the gaps spread over a wider interval as the size of instances grow. Similar plots could be obtained with the results of Benders-3*.

6 Conclusions

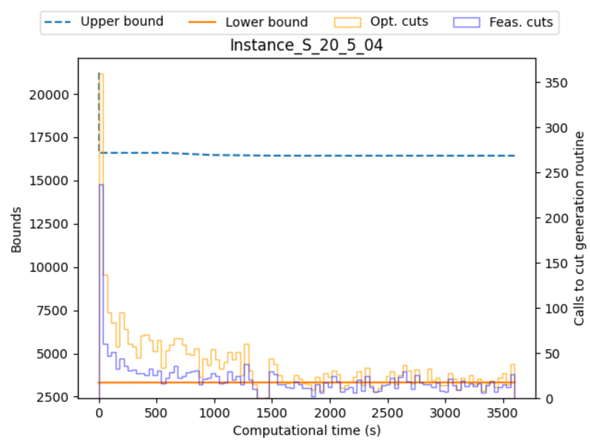
In this paper we have studied the TDVRP which is a very practical transportation problem. Due to the NP-hard nature of this problem, over the last decade, approximate solution procedures have been applied. This paper is the first to present an exact method using logic based Benders decomposition (LBBD) for this class of problems. Two formulations of the problem, based on the well-known two- and



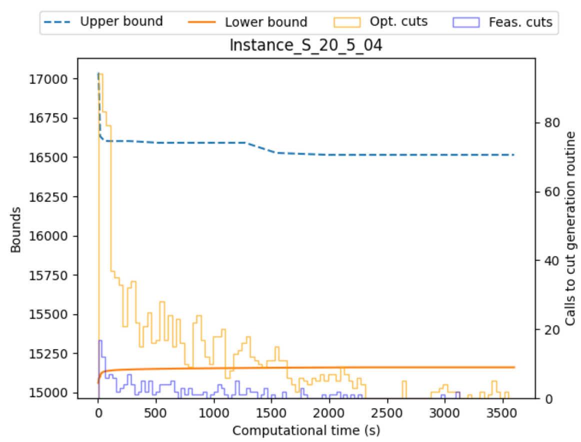
(a) Example 1 with Benders-2 (10 customers).



(b) Example 1 with Benders-3 (10 customers).



(c) Example 2 with Benders-2 (20 customers).



(d) Example 2 with Benders-3 (20 customers).

Figure 2: Profile of the evolution of bounds and cut generation routine calls during the solution process. For these figures, we aggregated length cuts into feasibility cuts.

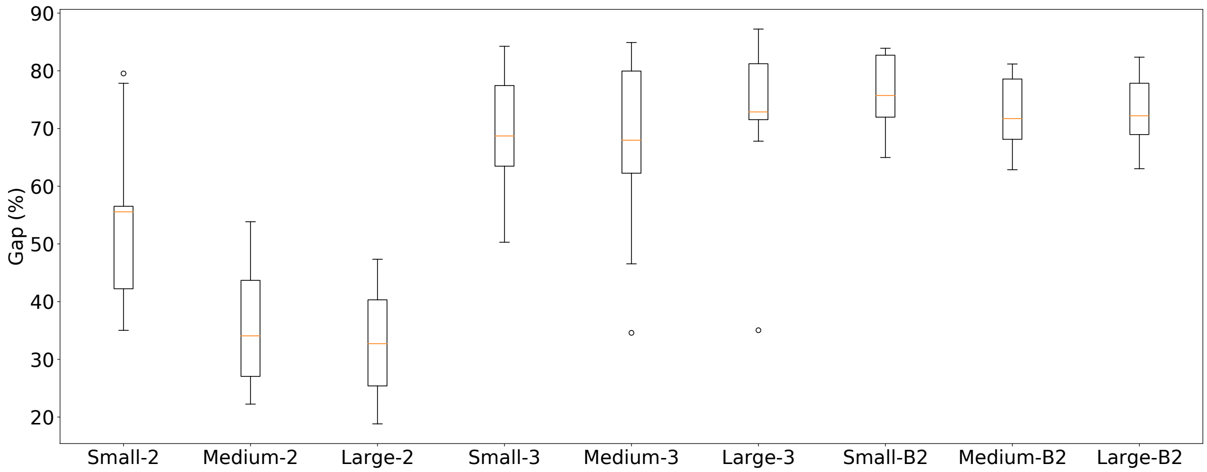


Figure 3: Box plot of gap for small, medium and large instances (with 10 customers) for Model-2 (Small-2, Medium-2 and Large-2), Model-3 (Small-3, Medium-3 and Large-3) and Benders-2 (Small-2B, Medium-2B and Large-2B) solution approaches, respectively.

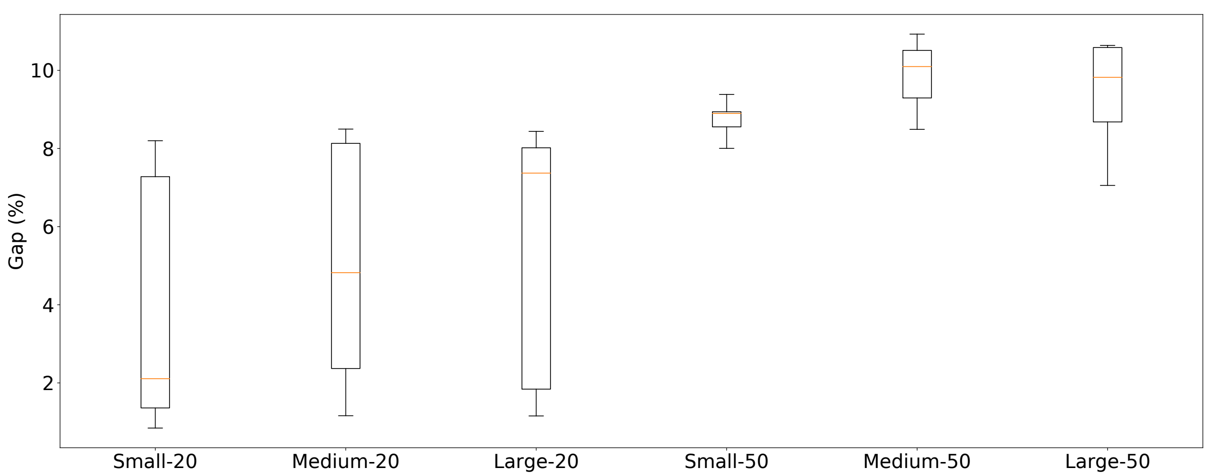


Figure 4: Box plot of gap for for small, medium and large instances for Benders-3 for instances with 20 and 50 customers, respectively.

three-index formulations for the VRP, have been proposed. The performance of the proposed Benders decomposition based method is compared against the results obtained by the branch-and-cut algorithm. Computational experiments on instances from the literature confirm that the proposed LBBD algorithm finds better solutions. However, the performance of the algorithm depends on the formulation used for the problem. The decomposition applied to the three-index formulation for the TDVRP has yielded significantly better solutions in less time due to the quality of the generated cuts. A comprehensive analysis of the results sheds light on the reasons for the superior performance of this method.

References

[1] B.-H. Ahn and J.-Y. Shin. Vehicle-routeing with time windows and time-varying congestion. *Journal of the Operational Research Society*, 42(5):393–400, 1991.

[2] F. Alkaabneh, A. Diabat, and H. O. Gao. Benders decomposition for the inventory vehicle routing problem with perishable products and environmental costs. *Computers & Operations Research*, 113:104751, 2020.

[3] J. F. Benders. Partitioning procedures for solving mixed-variable programming problems. *Numerische Matkematic*, 4(1):238–252, 1962.

[4] G. Codato and M. Fischetti. Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.

[5] S. Dabia, S. Ropke, T. Van Woensel, and T. De Kok. Branch and price for the time-dependent vehicle routing problem with time windows. *Transportation Science*, 47(3): 380–396, 2013.

[6] G. Dantzig, R. Fulkerson, and S. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, 2(4):393–410, 1954.

[7] M. Darvish, L. C. Coelho, and R. Jans. Comparison of symmetry breaking and input ordering techniques for routing problems. Technical report, CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Working Paper CIRRELT-2020-22, July 2020.

[8] A. V. Donati, R. Montemanni, N. Casagrande, A. E. Rizzoli, and L. M. Gambardella. Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research*, 185(3):1174–1191, 2008.

[9] R. F. Fachini and V. A. Armentano. Logic-based benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. *Computers & Industrial Engineering*, 148:106641, 2020.

[10] M.A. Figliozzi. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics. *Transportation Research*, 48(3):616–636, Mai. 2012.

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

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

[13] B. L. Golden, T. L Magnanti, and H. Q. Nguyen. Implementing vehicle routing algorithms. *Networks*, 7(2):113–148, 1977.

[14] A. Heching, J. N. Hooker, and R. Kimura. A logic-based benders approach to home healthcare delivery. *Transportation Science*, 53(2):510–522, 2019.

[15] J. Hooker. *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*, volume 2, chapter 19, pages 389–422. John Wiley & Sons, New York, 2011.

[16] J. N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

[17] Y. Huang, L. Zhao, T. Van Woensel, and J.-P. Gross. Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95: 169–195, 2017.

[18] 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.

[19] A.L. Kok, E.W. Hans, and J.M.J. Schutten. Vehicle routing under time-dependent travel times: the impact of congestion avoidance. *Computers & Operations Research*, 39(5):910–918, Mai. 2012.

[20] G. Laporte and Y. Nobert. Exact algorithms for the vehicle routing problem. In *North-Holland Mathematics Studies*, volume 132, pages 147–184. Elsevier, 1987.

[21] J. Li, D. Wang, and J. Zhang. Heterogeneous fixed fleet vehicle routing problem based on fuel and carbon emissions. *Journal of Cleaner Production*, 201:896–908, 2018.

[22] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445, 2004.

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

[24] S. Mancini. Time dependent travel speed vehicle routing and scheduling on a real road network: The case of torino. *Transportation Research Procedia*, 3:433–441, 2014.

[25] 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.

[26] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329, 1960.

[27] B. Naderi, K. Govindan, and H. Soleimani. A benders decomposition approach for a real case supply chain network design with capacity acquisition and transporter planning: wheat distribution network. *Annals of Operations Research*, 291(1):685–705, 2020.

[28] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3): 801–817, 2017.

[29] M. I Restrepo, B. Gendron, and L.-M. Rousseau. Combining benders decomposition and column generation for multi-activity tour scheduling. *Computers & Operations Research*, 93:151–165, 2018.

[30] M. Riedler and G. Raidl. Solving a selective dial-a-ride problem with logic-based benders decomposition. *Computers & Operations Research*, 96:30–54, 2018.

[31] C. E. Schmidt, A. C. L. Silva, M. Darvish, and L. C. Coelho. The time-dependent location-routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 128(C):293–315, 2019.

[32] D. Soler, J. Albiach, and E. Martínez. A way to optimally solve a time-dependent vehicle routing problem with time windows. *Operations Research Letters*, 37(1):37–42, 2009.

[33] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[34] E. S. Thorsteinsson. Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming. In T. Walsh, editor, *International Conference on Principles and Practice of Constraint Programming*, volume 2239 of Lecture Notes in Computer Science, pages 16–30, 2001.

[35] P. Toth and D. Vigo, editors. *Vehicle Routing.* Monographs on Discrete Mathematics and Applications. MOS-SIAM Series on Optimization, Philadelphia, 2014.

[36] 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.

[37] Thibaut Vidal, Gilbert Laporte, and Piotr Matl. A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*, 286(2):401–416, 2020.