Branch-Cut-and-Price for the Time-Dependent Green Vehicle Routing Problem with Time Windows

Branch-Cut-and-Price for the Time-Dependent Green Vehicle Routing Problem with Time Windows

Yiming Liu, Yang Yu
State Key Laboratory of Synthetic Automation for Process Industries, Institute of Intelligent Data and Systems Engineering,Northeastern University,Shenyang 110819, P.R. China yuyang@ise.neu.edu.cn,

Yu Zhang
School of Business Administration, Southwestern University of Finance and Economics,Chengdu 611130, P.R. China ,

Baldacci Roberto
College of Science and Engineering, Hamad Bin Khalifa University, Doha P.O. Box 5825, Qatar ,

Jiafu Tang
College of Management Science and Engineering,Dongbei University of Finance and Economics,Shahekou, Dalian, 116026, P.R. China ,

Xinggang Luo
School of Management,Hangzhou Dianzi University,Hangzhou, P.R. China ,

Wei Sun
Business School,Liaoning University,Shenyang 110316, P.R. China ,

Motivated by rising concerns regarding global warming and traffic congestion effects, we study the time-dependent green vehicle routing problem with time windows (TDGVRPTW), aiming to minimize carbon emissions. The TDGVRPTW is a variant of the time-dependent vehicle routing problem (TDVRP) in which, in addition to the time window constraints, the minimization of carbon emissions requires determination of the optimal departure times for vehicles, from both the depot and customer location(s). Accordingly, the first exact method based on a branch-cut-and-price (BCP) algorithm is proposed for solving the TDGVRPTW. We introduce the notation of a time-dependent (TD) arc, and describe how to identify the non-dominated TD arcs in terms of the arc departure times. In this way, we reduce infinitely many TD arcs to a finite set of non-dominated TD arcs. We design a state-of-the-art BCP algorithm for the TDGVRPTW with labeling and limited memory subset row cuts, together with effective dominance rules for eliminating dominated TD arcs. The exact method is tested on a set of test instances derived from benchmark instances proposed in the literature. The results show the effectiveness of the proposed exact method in solving TDGVRPTW instances involving up to 100 customers.

Key words: time-dependent; green vehicle routing problem; branch-cut-and-price.
History: March 18, 2022.

1. Introduction

The literature on the vehicle routing problem (VRP) and its variants is abundant (Laporte 2009, Baldacci et al. 2011, Vidal et al. 2014, Costa et al. 2019). In the era of global warming, there are opportunities to reduce carbon emissions by extending the traditional VRP objectives to account for environmental costs. The importance of green logistics is motivated by the fact that current logistics strategies are not sustainable in the long term (Lin et al. 2014). The ever-growing concern over greenhouse gases (GHG) has led many countries to take policy actions aiming at emissions reductions (Jabali et al. 2012). Despite improvements in implementing "green technology" to decrease GHG emissions, statistics show that the amount of pollutants (mainly the carbon dioxide equivalent) has increased (Moghdani et al. 2021). Increases in GHG emissions, especially carbon dioxide (CO_2), have led to global warming, which is expected to lead to far-reaching negative effects on the earth and human beings. An effective measure for mitigating these negative effects is to reduce the CO_2 emissions generated by the transportation sector. In the United States, the transportation sector contributes 29% of national GHG emissions, GHG emissions totaled 6,558 million metric tons of carbon dioxide equivalents (USEPA 2021). A growing interest toward "green transportation" and growing concerns about such hazardous effects of transportation on the environment call for revised planning approaches to road transportation by explicitly accounting for such negative impacts (Bektaş and Laporte 2011, Andelmin and Bartolini 2017). Motivated by the practical importance of these problems, green VRPs (GVRPs) and its variants have been investigated by several authors (Bektaş and Laporte 2011, Demir et al. 2014a,b, Bektaş et al. 2016, Yu et al. 2016, Wang et al. 2018, Yu et al. 2019a,b, Sun et al. 2019). In a GVRP, the environmental, ecological, and social effects are considered when designing logistics policies, in addition to the conventional economic costs (Lin et al. 2014). As an additional matter, in real contexts (such as in dense urban areas), factors such as traffic jams can render vehicle travel times highly dependent on time. Ignoring the time dependency of travel times can result in suboptimal solutions; this has motivated studies on time-dependent GVRPs (TDGVRPs), where the travel time and carbon emissions between two locations depend on the time of the day (Franceschetti et al. 2013, Desaulniers et al. 2014, Franceschetti et al. 2017, Çimen and Soysal 2017, Kazemian et al. 2018). Solving TDGVRPs to optimality is challenging. The vehicle speeds in TDGVRPs are time-dependent, so that the carbon emissions along the routes depend on the departure times from the depot and customer locations. Time-dependent VRPs (TDVRPs) typically minimize the total travel duration, i.e., a linear function, whereas TDGVRPs minimize the total carbon emissions, i.e., a nonlinear function. Moreover, in TDGVRPs, the departure times from the *customers* also matter, as waiting may cause less carbon emissions. The additional dimensions of these decision variables enlarge the solution space, and deserve a careful study of efficient solution approaches for TDGVRPs.

Motivated by the practical importance of the problem, in this paper we study the TDGVRPTW to minimize carbon emissions, and we present an exact algorithm for its solution.

1.1. Related work

As a member of the VRP family, the TDGVRP with time windows (TDGVRPTW) has prominent features concerning time dependence in regards to the travel times, and non-linearity in the carbon emission functions. This section reviews the related literature on TDVRPs, GVRPs, and TDGVRPs.

There is vast literature on VRPs, and many (meta-)heuristics and exact methods have been proposed for solving different VRP variants (see, for example, Kelly and Xu (1999), Bräysy (2003), Baldacci et al. (2004, 2012), Costa et al. (2019)). Toth and Vigo (2014) provides a comprehensive overview of the exact and heuristic methods for VRPs.

Time-dependent Vehicle Routing Problem Based on the First-In-First-Out (FIFO) property in the TDVRP, Dabia et al. (2013) exploited the fact that optimal routes bear no waiting at customer nodes, and proposed an effective functional dominance rule for determining the optimal departure times from the *depot*. The FIFO property guarantees that a vehicle leaving a node i at any time t to go to the next node j cannot arrive earlier than another vehicle leaving i before t to go directly to j (Ichoua et al. 2003, Desaulniers et al. 2014). Malandraki and Dial (1996) proposed a restricted dynamic programming heuristic (a generalization of the nearest neighbor heuristic) for solving the time-dependent traveling salesman problem (TDTSP). Vu et al. (2020) presented a new approach based on an integer programming formulation of the TDTSP with time windows (TDTSPTW) over a time-expanded network. In this network, all departure times from nodes were discretized, such that the time-expanded network could have a finite set of nodes and a finite set of arcs. For the TDVRP, most solution approaches are based on (meta-)heuristics. Malandraki and

Daskin (1992) presented a mixed-integer linear formulation for the TDVRP, and proposed a heuristic algorithm based on the nearest neighbor. Ichoua et al. (2003) stated that the TDVRP has FIFO properties, and proposed a parallel tabu search heuristic algorithm. Donati et al. (2008) proposed an ant colony optimization algorithm for solving the TDVRP with time windows (TDVRPTW), whereas Hashimoto et al. (2008) presented an iterated local search algorithm. Pan et al. (2021) designed a hybrid meta-heuristic algorithm for solving a multi-trip TDVRPTW (MT-TDVRPTW). There are two exact algorithms for the TDVRP in the literature. Soler et al. (2009) proposed auxiliary digraphs, and transformed the TDVRPTW into an asymmetric capacitated VRP by discretizing the departure times from the depot and customer locations. Dabia et al. (2013) presented a branch-and-price (BP) algorithm for the TDVRPTW, aiming to minimize the total route duration. Owing to the FIFO property, only the route departure times from the depot needed to be determined. Evidently, because traffic conditions are greatly affected by urban environments, considering the time dependence makes the solutions of TDVRPs more realistic.

Green Vehicle Routing Problem Bektaş and Laporte (2011) presented a pollution-routing problem (PRP) with an objective function accounting for not only the travel distance, but also the amount of greenhouse emissions, fuel, and travel times, and their costs. Demir et al. (2014a) investigated a bi-objective PRP, and used an adaptive large-neighborhood search algorithm to solve it. Fukasawa et al. (2016) proposed a BCP algorithm for an energy-minimizing VRP. Dabia et al. (2017) presented a BP algorithm for a variant of the PRP, in which the speed and start time at the depot needs to be decided on for each individual route. Yu et al. (2019a) presented a BP algorithm for a heterogeneous fleet green VRP with time windows. Yu et al. (2019b) investigated a bi-objective green ride-sharing problem, and used an exact method to solve it. However, they assumed that the vehicle speed was fixed, which is not realistic for most practical applications. Because vehicles' carbon emissions are related to their speed, a GVRP considering time dependence could be more realistic.

Table 1 **Overview of time-dependent vehicle routing problems (TDVRPs) and time-dependent green vehicle routing problems (TDGVRPs)**

References	**Problem features**	**Algorithmic features**	**Objectives**	**Data set**
Malandraki and Dial (1996)	TDTSP	Nearest neighbor heuristic	Min. return time to the depot	Randomly generated problems
Vu et al. (2020)	TDTSPTW	Integer programming formulation	Min. makespan and duration	Benchmark Results
Malandraki and Daskin (1992)	TDTSP and TDVRP	Mixed-integer linear formulation; Nearest neighbor heuristic	Min. total route time	Randomly generated problems
Ichoua et al. (2003)	TDVRPTW	Parallel tabu search heuristic	Min. total travel time	Solomon's benchmark data
Donati et al. (2008)	TDVRPTW	Ant colony optimization algorithm	Min. tour and total length considering traveling time	Solomon's benchmark data
Hashimoto et al. (2008)	TDVRPTW	Iterated local search	Min. travel time and travel cost	Solomon's benchmark data
Pan et al. (2021)	MT-TDVRPTW	Hybrid metaheuristic	Min. the total travel distance	Solomon's benchmark data
Soler et al. (2009)	TDVRPTW	Auxiliary digraphs	Min. travel cost	No computation result
Dabia et al. (2013)	TDVRPTW	Branch-and-price	Min. duration	Solomon's benchmark data
Franceschetti et al. (2013)	TDPRP	Integer linear programming formulation and departure time and speed optimization	Min. carbon emissions, travel times and drivers costs	Three sets of instances from the PRPLIB
Dabia et al. (2017)	PRP with same speed on each arc and the determination of depot departure time	Branch-and-price	Min. fuel and driving costs	Solomon's benchmark data
Franceschetti et al. (2017)	TDPRP	Adaptive large neighborhood search heuristic	Min. carbon emissions, travel times and drivers costs	Sets of instances from the PRPLIB
Çimen and Soysal (2017)	TDGVRP with stochastic vehicle speeds	Approximate dynamic programming	Min. carbon emissions	Sets of instances from the PRPLIB
Kazemian et al. (2018)	VRP with loading cost (VRPLC)	Auxiliary digraphs	Min. loading cost	Randomly generated problems

Time-dependent GVRP Kuo (2010) used simulated annealing to minimize the fuel consumption for the TDVRP. Franceschetti et al. (2013) provided an analytical characterization of the optimal solution for the time-dependent PRP (TDPRP) that minimized carbon emissions, travel times, and driver costs. Franceschetti et al. (2017) proposed an adaptive large neighborhood search heuristic for solving the TDPRP. Çimen and Soysal (2017) proposed an approximate dynamic programming method based on a heuristic as a decision aid tool for solving the TDGVRP with stochastic vehicle speeds. Kazemian et al. (2018) transformed the TDGVRPTW into a VRP with loading costs (VRPLC); they then used the existing algorithms for the VRPLC to solve it.

Table 1 lists the key features of the works proposed in the literature for TDVRPs and TDGVRPs. To the best of our knowledge, no exact approach has been proposed in the literature for the TDGVRPTW.

1.2. Our contributions

In this study, we present the first exact method based on a branch-cut-and-price (BCP) algorithm for solving the TDGVRPTW. The main contributions of this study are summarized as follows.

- We introduce the notion of a non-dominated time-dependent (TD) arc based on a time-expended network. This has the benefit of considering only a finite subset of departure times from customer nodes, thereby reducing the solution space. We then prove that an optimal timed route comprises a sequence of non-dominated TD arcs, and formulate the problem using a set-partitioning model.
- We describe methods for determining the non-dominated TD arcs, i.e., given an arc, we determine the optimal departure time from the allowable time interval. In some situations, we prove that the earliest allowable departure times are optimal, and in other cases, we discretize the allowable departure times and determine the optimal departure times. In this way, we reduce infinitely many TD arcs to a finite set of non-dominated TD arcs.
- We design a state-of-the-art BCP algorithm for the TDGVRPTW with labeling and limited memory subset row cuts (lm-SRCs), together with effective dominance rules for eliminating dominated TD arcs.
- We perform extensive computational experiments, and the results show the effectiveness of the proposed solution method.

1.3. Overview of the paper

The remainder of this paper proceeds as follows. Section 2 describes the TDGVRPTW. According to the characteristics of the TDGVRPTW, the optimal timed route and the non-dominated TD arc are defined based on a time-expanded network to reduce the solution space. We then present the relationship between carbon emissions and departure times, followed by the procedure used to identify the non-dominated TD arcs. Section 3 introduces the mathematical formulation and valid inequalities used to solve the TDGVRPTW. Section 4 develops a BCP algorithm strengthened with lm-SRCs, and a pricing algorithm based on embedding new dominance rules to eliminate unpromising timed routes. Section 5 reports extensive computational experiments on benchmark instances from the literature. The conclusions are presented in Section 6.

2. Time-dependent green vehicle routing problem with time windows (TDGVRPTW) and non-dominated Time-dependent (TD) arcs

In this section, we first introduce the TDGVRPTW followed by the description of the time expanded network used to model TDGVRPTW solutions and to identify non-dominated TD arcs.

2.1. Problem description

We describe the TDGVRPTW by first describing the GVRPTW and then by considering the time-dependent travel times.

The GVRPTW aims to find a set of delivery routes visiting each customer exactly once, so as to minimize the total carbon emissions while satisfying the demand and time window for each customer. Consider a network $G = (V, A)$, where the set of nodes $V = \{0, 1, \ldots, n+1\}$ comprises starting depot 0, set $N = \{1, 2, \ldots, n\}$ of customer nodes, and ending depot $n+1$; the set of arcs is defined as $A = \{(0, j)|j \in N\} \cup \{(i, j)|i, j \in N, i \neq j\} \cup \{(i, n+1)|i \in N\}$. Let $[e_i, l_i]$ be the time window where e_i and l_i represent the prescribed earliest and latest times at which service can start at customer i, respectively; q_i is the demand; and s_i is the service time of node $i \in V$. Without loss of generality, we assume that $s_0 = s_{n+1} = q_0 = q_{n+1} = 0$. The total carbon emissions c_{ij} across arc $(i, j) \in A$ are computed based on the definitions given by Bektaş and Laporte (2011) and Franceschetti et al. (2013), as follows:

$$c_{ij} = \gamma f P_{ij} = \gamma f[\alpha_{ij}(W + w_{ij})d_{ij} + \beta v_{ij}^2 d_{ij}], \tag{1}$$

where γ is the carbon emissions index parameter; f is the fuel consumption index parameter; P_{ij} is the total amount of energy consumed on arc $(i,j) \in A$; $\alpha_{ij} = a + gsin\theta_{ij} + gC_r cos\theta_{ij}$ is an arc specific constant; $\beta = 0.5C_d A\rho$ is a vehicle specific constant; W is the empty vehicle weight; w_{ij} is the vehicle load on arc $(i,j) \in A$; v_{ij} is the vehicle speed on arc (i,j); d_{ij} is the distance of arc (i,j); a is the acceleration (m/s^2); g is the gravitational constant (m/s^2); θ_{ij} is the arc angle; C_r is the coefficients of rolling resistance; C_d is the drag coefficient; A is the frontal surface area of the vehicle (m^2); and ρ is the air density (kg/m^3).

For the TDGVRPTW, owing to the time dependence of the vehicle speed, the speed of the vehicle on arc $(i,j) \in A$ may change over time. Thus, the travel time and carbon emissions between the two nodes depend on the departure time from node i. Figure 1 shows an example of a vehicle speed profile associated with a TD case.

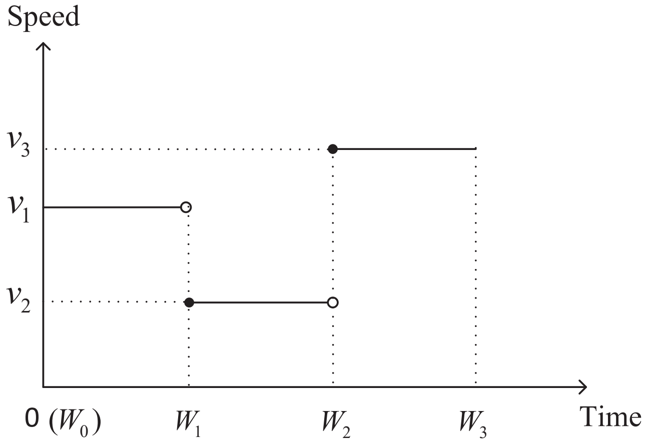

Figure 1 **Example of vehicle speed changing over time**

In Figure 1, $[0(W_0), W_3]$ represents the planning horizon; W_1, W_2 and W_3 are speed breakpoints where speeds change; vehicle speeds are v_1, v_2, and v_3 in $[W_0, W_1)$, $[W_1, W_2)$, and $[W_2, W_3]$ respectively. From the figure, the vehicle leaves node $i \in V$ to node $j \in V$ at departure time t_i in $[W_0, W_1)$ and travels at speed v_1 until it reaches the time point W_1. From then, the vehicle travels at a slower speed v_2 until W_2. Finally, it travels at speed v_3 for the remaining time interval (Ichoua et al. 2003).

According to Figure 1, the travel time $\tau_{ij}(t_i)$ across arc $(i,j) \in A$ can be calculated as follows:

$$\tau_{ij}(t_i) = \sum_{k=m}^{k=m+h} \frac{d_k}{v_k}, \tag{2}$$

In the above, v_k is the vehicle speed traveling from $i \in V$ to $j \in V$ in the time interval $[W_{k-1}, W_k]$; d_k is the traveling distance that vehicle speed is v_k; it holds that $\sum_{k=m}^{k=m+h} d_k = d_{ij}$; m is the index of the time interval $[W_{m-1}, W_m]$ that includes t_i (i.e., $m \mid t_i \in [W_{m-1}, W_m]$), and is known for a given t_i; and h is the number of changes in the vehicle speed in arc (i,j). The function for the travel time across arc $(i,j) \in A$ was first proposed by Ichoua et al. (2003), and was also used by Kuo (2010) and Franceschetti et al. (2013).

As the vehicle speed on arc $(i,j) \in A$ may change, the carbon emissions cannot be computed using Eq. (1). Indeed, Bektaş and Laporte (2011) computed carbon emissions as follows:

$$c_{ij}(t_i) = \gamma f[\alpha_{ij}(W + w_{ij})d_{ij} + \beta \sum_{k=m}^{k=m+h} v_k^2 d_k]. \tag{3}$$

Based on Eq. (3), given a departure time t_i, the carbon emissions across arc $(i,j) \in A$ can be calculated. Moreover, Eq. (3) indicates that different departure times may cause different carbon emissions; therefore, the departure time from each node must be determined.

The TDGVRPTW studied herein differs from the TDVRPTW, which minimizes the travel duration. The TDVRPTW needs to find a set of delivery routes and determine the departure time from the depot, whereas the TDGVRPTW needs to determine the departure times from customer nodes along each route. In the TDVRPTW, the FIFO property ensures that, given a depot departure time, a route without waiting at customer nodes has the shortest travel duration. However, in the TDGVRPTW, the departure times from customer nodes matter - they affect the carbon emissions, and thus are essentially decision variables. When the goal is to minimize the travel time (or the carbon emissions in our study), it may be beneficial to wait at any location (Boland and Savelsbergh 2019). This new feature dramatically enlarges the solution space, and requires a careful design of solution algorithms.

2.2. Modeling arcs and routes on a time-expanded network

Time-indexed (TI) formulations are effective formulations for solving TDVRPs and their variants. TI formulations require a discretization of time fine enough to provide a correct model; providing a complete TI model for the TDTSPTW with an objective where it may be beneficial to wait at any location (as in the TDGVRPTW) is not straightforward (Boland and Savelsbergh 2019). Nevertheless, in several cases, a fine discretization yields optimal or near-optimal solution. For this reason, we model the problem on a time-expanded network, and investigate ways to reduce its dimensions by investigating the properties of the optimal timed routes.

We consider a time-expanded network $\mathcal{D}=(\mathcal{N},\mathcal{B})$ with a node set $\mathcal{N}$ and arc set $\mathcal{B}$ (Vu et al. 2020, Boland and Savelsbergh 2019). Let $\mathcal{N}$ contain timed node (i,t_i) for $i\in V$ and $t_i\in[ed_{ij},ld_{ij}]$, and $[ed_{ij},ld_{ij}]$ is the feasible departure time interval when traversing arc $(i,j)\in A$, where ed_{ij} and ld_{ij} represent the earliest and latest feasible departure times, respectively. Furthermore, let $\mathcal{B}$ contain arcs of the form $((i,t_i),(j,t_j^a))$ with $i\neq j$, $(i,j)\in A$, $t_i\in[ed_{ij},ld_{ij}]$, and arrival time $t_j^a=t_i+\tau_{ij}(t_i)$. We call this form of the arc *time-dependent arc* (*TD arc*). To identify the TD arcs, for each $((i,t_i),(j,t_j^a))$, we define $A_{ij}\subset\mathcal{B}$ as the set of all TD arcs corresponding to arc $(i,j)\in A$, i.e., $A_{ij}=\{((i,t_i),(j,t_j^a))\mid t_i\in[ed_{ij},ld_{ij}]\}$. The value $c_{ij}(t_i)$ represents the carbon emissions across the TD arc $((i,t_i),(j,t_j^a))\in A_{ij}$. In contrast to the TDTSPTW, which only determines the departure time in the time window of each node (Vu et al. 2020), in the TDGVRPTW, the vehicle can leave the location of customer i later than the latest time l_i of the time window $[e_i,l_i]$. As waiting at each node (depot and customer) may lead to reduced carbon emissions, we define two types of waiting time to distinguish the difference between them, and in Section 5.6, we present the benefits of waiting after finishing service.

Definition 1 (Type 1 waiting time of customer j, $Type1\_wt(j)$). Waiting time before the earliest service time of customer j.

Let $[e_j,l_j]$ be the time window of node j, and t_j^a be the arrival time of j. The Type 1 waiting time of customer j can be calculated as follows:

$$Type1\_wt(j)=\begin{cases} e_j-t_j^a, & \text{if } t_j^a<e_j,\\ 0 & otherwise.\end{cases}\tag{4}$$

Definition 2 (Type 2 waiting time of customer j, $Type2\_wt(j)$). Waiting time after service of customer j.

Let t_j be the departure time of j, and fs_j be the service finishing time of j. The Type 2 waiting time of customer j can be calculated as follows:

$$Type2\_wt(j) = t_j - fs_j. \tag{5}$$

We define the node sequence of a timed route as $S = (0, i_1, ..., i_z, n+1)$ in G with $i_1, \ldots, i_z \in N$. Associated with the node sequence S are the different departure times from the depot and customer locations. Therefore, let Ω_S be the set of all feasible timed routes corresponding to node sequence S; and $S_r \in \Omega_S$ is the r^{th} feasible timed route corresponding to the node sequence S. For the node sequence $S = (0, i_1, ..., i_z, n+1)$, let $S_r = ((0, t_0), (i_1, t_{i_1}), \ldots, (i_z, t_{i_z}), (n+1))$ be the timed route, where t_{i_m} is the departure time at node i_m for all $m = 1, 2, \ldots, z$. Let c_{S_r} denote the carbon emissions of S_r.

The cardinality of set Ω_S is large, but its cardinality can be reduced by means of the following definition of the *optimal timed route*.

Definition 3 (Optimal timed route of S). S_r is the optimal timed route of node sequence $S = (0, i_1, \ldots, i_z, n+1)$ if and only if $c_{S_r} \le c_{S_{r'}}$, $\forall S_{r'} \in \Omega_S$.

Definition 3 indicates that the optimal timed route for S has the minimal carbon emissions. However, there may be more than one timed route with minimal carbon emissions. In this case, the timed route with the earliest departure time for each node is defined as the timed optimal route, and thus the optimal timed route can be obtained using lexicographical order. A simple example is given in Appendix A of the e-companion to this paper.

2.3. Non-dominated TD arcs

To eliminate non-optimal timed routes in the TDGVRPTW, in this section, we define a non-dominated TD arc, and prove that the *optimal* timed route is composed of non-dominated TD arcs.

Definition 4 (Dominance relation of TD arc of A_{ij}, $\succ_{TDarc_{ij}}$). Let $((i, t_i), (j, t_j^a)), ((i, t_i^{'}), (j, t_j^{a^{'}})) \in A_{ij}$ with $i, j \in V$. Arc $((i, t_i), (j, t_j^a))$ dominates arc $((i, t_i^{'}), (j, t_j^{a^{'}}))$ (i.e., $((i, t_i), (j, t_j^a)) \succ_{TDarc_{ij}} ((i, t_i^{'}), (j, t_j^{a^{'}})))$ if $t_i \le t_i^{'}$ and $c_{ij}(t_i) \le c_{ij}(t_i^{'})$, where at least one inequality is strict.

Definition 5 (Non-dominated TD arc of A_{ij}). Arc $((i,t_i),(j,t_j^a)) \in A_{ij}$ is a non-dominated TD arc if $\nexists((i,t_i^{'}),(j,t_j^{a^{'}})) \in A_{ij}$ such that $((i,t_i^{'}),(j,t_j^{a^{'}})) \succ_{TDarc_{ij}} ((i,t_i),(j,t_j^a))$, with $i,j \in V$.

Based on the above definition, the following theorem holds.

Theorem 1. *An optimal timed route is composed of non-dominated TD arcs.*

Proof. See Appendix B of the e-companion.

Theorem 1 indicates that all dominated TD arcs can be eliminated to obtain the optimal timed routes for the TDGVRPTW. Because an arbitrary optimal timed route is composed of non-dominated TD arcs, we develop an approach for identifying non-dominated TD arcs.

2.4. Identifying non-dominated TD arcs

In this section, we present a method for identifying non-dominated TD arcs. We first describe the relationship between carbon emissions and departure times, and then present a method for identifying non-dominated TD arcs.

2.4.1. Relationship between the carbon emissions and departure time of arc (i,j).

As shown in Eq. (3), when the distance d_{ij} and vehicle load w_{ij} are given, the carbon emissions of arc $(i,j) \in A$ depend on the departure time of node $i \in V$. For the speed profile shown in Figure 1, the relationship between the carbon emissions and departure times of arc (i,j) is shown in Figure 2.

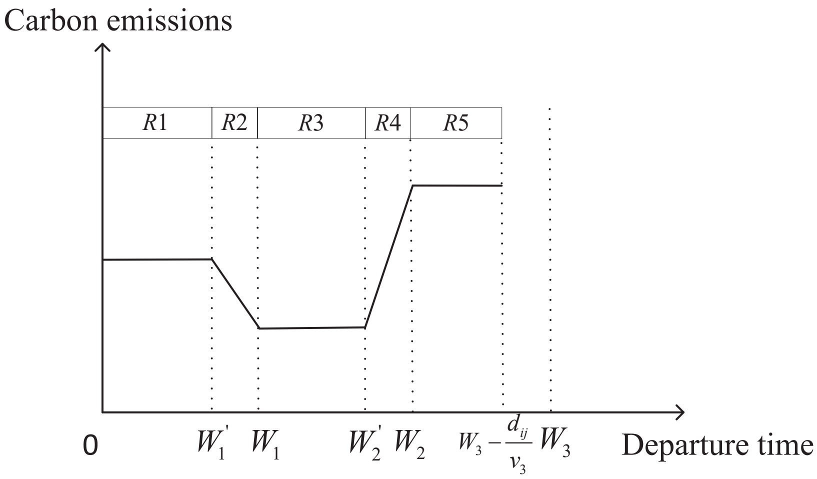

Figure 2 **Carbon emissions function of arc** (i,j)

In Figure 2, d_{ij} is the distance of arc (i,j). W_1, W_2 and W_3 are speed breakpoints in Figure 1 where speeds change. Also, W_1, W_2 and W_3 are carbon emissions breakpoints where carbon emissions change. The other carbon emissions breakpoints (i.e., $W_1^{'}$ and $W_2^{'}$) are departure times such that the arrival times are exactly the speed breakpoints (e.g., $W_1^{'} = W_1 - d_{ij}/v_1$ is the departure time at node i so as to arrive at time W_1 at node j).

In addition, the slope of the function of $[W_1^{'}, W_1]$ is $\gamma f \beta (v_1 v_2^2 - v_1^3)$, and the slope of the function of $[W_2^{'}, W_2]$ is $\gamma f \beta (v_2 v_3^2 - v_2^3)$.

Based on Figure 2, for a general speed profile (i.e., see Figure EC.2 in Appendix C of the e-companion), the slope of the carbon emissions function (i.e., see Figure EC.3 in Appendix C of the e-companion) that corresponds to the general speed profile can be calculated as $\gamma f \beta (v_k v_{k+1}^2 - v_k^3)$, where v_k is the vehicle speed traveling from $i \in V$ to $j \in V$ in an arbitrary time interval $[W_k^{'}, W_k]$, and where $W_k^{'}$ and W_k are adjacent breakpoints. The detailed derivations of the slopes are provided in Appendix C of the e-companion.

2.4.2. Planning horizon division and identification of non-dominated TD arcs. As shown in Figure 2, the carbon emissions function is piecewise linear. In the case of the TDGVRPTW, the situation is more complicated. Thus, we divide the planning horizon $[0, W_3]$ into five regions: Region 1 (*R1*) $=[0, W_1^{'}]$; Region 2 (*R2*) $=[W_1^{'}, W_1]$; Region 3 (*R3*) $=[W_1, W_2^{'}]$; Region 4 (*R4*) $=[W_2^{'}, W_2]$; and Region 5 (*R5*) $=[W_2, W_3 - d_{ij}/v_3]$. Subsequently, by investigating the characteristic of each region, we observe that the carbon emissions of arc $(i,j) \in A$ are independent of the departure times when the departure time horizon (i.e.,$[ed_{ij}, ld_{ij}]$) is in certain regions.

Region 1. When $[ed_{ij}, ld_{ij}]$ is in *R1* (i.e., $[ed_{ij}, ld_{ij}] \subseteq [0, W_1^{'}]$), the vehicle speed is always v_1 from i to j. Therefore, the carbon emissions $c_{ij}(t_i)$ are time-independent of the departure time $t_i \in [ed_{ij}, ld_{ij}]$. According to the definition of the non-dominated TD arc (see Definition 5), the TD arc with the earliest departure time (i.e., $t_i = ed_{ij}$) is the only non-dominated TD arc. Thus, only arc $((i, ed_{ij}), (j, t_j^a))$ is considered, and the other TD arcs are eliminated.

Region 2. When $[ed_{ij}, ld_{ij}]$ is in *R2* (i.e., $[ed_{ij}, ld_{ij}] \subseteq [W_1^{'}, W_1]$), a TD arc with an earlier departure time $t_i \in [ed_{ij}, l_{ij}]$ has greater carbon emissions $c_{ij}(t_i)$. Thus, each TD arc is a non-dominated TD arc.

Region 3. Similar to *R1*, $((i, ed_{ij}), (j, t_j^a))$ is the only non-dominated TD arc.

Region 4. When $[ed_{ij}, ld_{ij}]$ is in *R4* (i.e., $[ed_{ij}, ld_{ij}] \subseteq [W_2^{'}, W_2]$), the TD arc with the earliest departure time (i.e., $t_i = ed_{ij}$) has the lowest carbon emissions $c_{ij}(t_i)$. Thus, the

TD arc with the earliest departure time ed_{ij} is considered as the only non-dominated TD arc.

Region 5. Similar to *R1*, $((i,ed_{ij}),(j,t_j^a))$ is the only non-dominated TD arc.

Based on the above description, for a general speed profile, the arc carbon emissions functions with time can be classified into three types. 1) arc carbon emissions functions independent with time (e.g., *Region 1*, *Region 3*, *Region 5*); 2) arc carbon emissions functions decreasing with time (e.g., *Region 2*); and 3) arc carbon emissions functions increasing with time (e.g., *Region 4*).

Based on the above classification, we identify the situations in which the carbon emissions of arc (i,j) are time-independent, as stated in the following theorem.

THEOREM 2. *The TD arc* $((i,t_i),(j,t_j^a))$ *with the earliest departure time (i.e.,* $((i,ed_{ij}),(j,t_j^a))$*) is the only non-dominated TD arc if the carbon emissions functions of* $((i,t_i),(j,t_j^a))$ *are independent of the departure time or increasing with the departure time (e.g.,* $[ed_{ij},ld_{ij}]\cap[W_1',W_1]=\emptyset$*).*

Proof. See Appendix D of the e-companion.

Theorem 2 states that when the carbon emissions functions of $((i,t_i),(j,t_j^a))$ are independent of the departure time or increasing with the departure time, the arc $((i,t_i),(j,t_j^a))$ with the earliest departure time is the only non-dominated TD arc. Hence, the following theorem holds.

THEOREM 3. *Each TD arc* $((i,t_i),(j,t_j^a))$ *is a non-dominated TD arc if the carbon emissions functions of* $((i,t_i),(j,t_j^a))$ *are decreasing with the departure time (e.g.,* $t_i\in[ed_{ij},ld_{ij}]\cap[W_1',W_1]\neq\emptyset$*).*

Proof. See Appendix E of the e-companion.

Theorem 3 indicates that if the departure time is continuous in the region in which the carbon emissions functions decrease with the departure time (e.g., $t_i\in[ed_{ij},ld_{ij}]\cap[W_1',W_1]\neq\emptyset$, the number of non-dominated TD arcs is infinite. To address this issue, we discretize the departure time in this region (e.g., $t_i\in[ed_{ij},ld_{ij}]\cap[W_1',W_1]\neq\emptyset$). The discretization scheme represents an approximation of real-world conditions and is commonly used in related literature (Ichoua et al. 2003, Vu et al. 2020).

2.4.3. Identifying non-dominated TD arcs under the carbon emissions decreasing with departure time. When the carbon emissions functions decrease with the departure time (e.g., $[l,u] = [ed_{ij}, ld_{ij}] \cap [W_1^{'}, W_1] \neq \emptyset$), each TD arc with departure time t_i in the region is non-dominated. To address the issue, we discretize the departure times in this region to $\lfloor u^{'} \rfloor - \lceil l^{'} \rceil + 1$ units, where $u^{'}$ and $l^{'}$ represent the upper and lower bounds of this region, respectively. The m^{th} discrete departure time in this region is $\lceil l^{'} \rceil + m - 1$. Because a 24-hour system is used to modify the time of benchmark instances, as described in Dabia et al. (2017), l and u are first converted to the original values (i.e., $l^{'} = l\frac{l_0^{'}}{24}$ and $u^{'} = u\frac{l_0^{'}}{24}$). Then, all the discrete departure times are obtained and converted back to the 24-hour system.

An example of a discrete departure time t_i in $[ed_{ij}, ld_{ij}] \cap [W_1^{'}, W_1] = [5.9, 8.2]$ of arc $(i,j) \in A$ is shown in Figure 3.

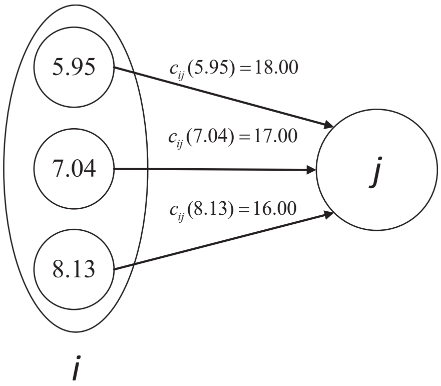

Figure 3 **Example of discrete departure times of arc** (i,j)

In Figure 3, there are three discrete departure times in $[5.9, 8.2]$, and these departure times are 5.95, 7.04 and 8.13. The first discrete departure time of node $i \in V$ is 5.95 and $c_{ij}(5.95) = 18.00$ is the carbon emissions with the departure time of 5.95 ($c_{ij}(7.04)$ and $c_{ij}(8.13)$ are similar to $c_{ij}(5.95)$). According to the definition of the non-dominated TD arc, all of three TD arcs are non-dominated TD arcs.

3. Mathematical formulation and valid inequalities

In this section, based on the definition of *optimal* timed routes, we describe a set partitioning formulation for the TDGVRPTW. Also in this section, we show how to strengthen the linear programming (LP)-relaxation of the set partitioning formulation by means of valid inequalities, and we describe the pricing problem associated with the formulation.

3.1. Set-partitioning formulation of the TDGVRPTW

Let Ω be the set of all the *optimal* timed routes. The binary parameter a_{iS_r} is equal to one if customer $i \in N$ is visited by timed route $S_r \in \Omega$, and zero otherwise. For each timed route $S_r \in \Omega$, we define a binary variable x_{S_r}, which is equal to 1 if S_r is selected in the solution and 0 otherwise. The set-partitioning model for the TDGVRPTW is as follows:

$$\min \sum_{S_r \in \Omega} c_{S_r} x_{S_r} \tag{6}$$

$$s.t. \sum_{S_r \in \Omega} a_{iS_r} x_{S_r} = 1, \, \forall i \in N \tag{7}$$

$$x_{S_r} \in \{0,1\}, \forall S_r \in \Omega. \tag{8}$$

As shown above, Function (6) aims to minimize the carbon emissions, Constraint (7) ensures that each customer is visited exactly once, and Constraint (8) defines the domain of the variables.

The LP-relaxation of the above formulation can be solved for using column generation techniques (Lübbecke and Desrosiers 2005), i.e., by iteratively solving a restricted master problem (RMP) defined by the above formulation over a restricted set of timed routes $\Omega' \subset \Omega$.

3.2. Valid inequalities

The master problem (MP) is a set-partitioning problem, and it is common practice to strengthen the MP with valid inequalities (Jepsen et al. 2008, Costa et al. 2019), such as the SRCs proposed by Jepsen et al. (2008), as follows:

$$\sum_{S_r \in \Omega} \left\lfloor \frac{1}{k} \sum_{i \in N'} a_{iS_r} \right\rfloor x_{S_r} \leq \left\lfloor \frac{|N'|}{k} \right\rfloor \tag{9}$$

where $N' \subseteq N$ and $0 < k \leq |N'|$.

We use lm-SRCs (Pecin et al. 2017b) instead of SRCs, which are defined as follows:

$$\sum_{S_r \in \Omega} \alpha(N', M, k, S_r) x_{S_r} \leq \left\lfloor \frac{|N'|}{k} \right\rfloor \tag{10}$$

where α is computed as a function of $(N^{'}, M, k, S_r)$ and satisfies $\alpha(N^{'}, M, k, S_r) \leq \left\lfloor \frac{1}{k} \sum_{i \in N'} a_{iS_r} \right\rfloor$; M is a memory set and $N^{'} \subseteq M \subseteq N$. The memory set M is constructed according to the function used by Pecin et al. (2017b).

The lm-SRCs can reduce the impact on the pricing algorithm used in the pricing problem relative to SRCs (Pecin et al. 2017b, Jepsen et al. 2008). We add (10) into the RMP when we use the lm-SRCs (see Section 4.3 for the corresponding details).

3.3. Pricing problem

The aim of the pricing problem in the TDGVRPTW is to search for the optimal timed routes with negative reduced costs, and to add these timed routes to the RMP. The reduced cost of a timed route can be computed as follows:

$$\bar{c}_{S_r} = c_{S_r} - \sum_{i \in N} a_{iS_r} \pi_i, \quad S_r \in \Omega, \tag{11}$$

Here, π_i is the dual variable associated with Constraint (7) for customer $i \in N$. The objective of the pricing problem is as follows:

$$\min \bar{c}_{S_r} = \min\{c_{S_r} - \sum_{i \in N} a_{iS_r} \pi_i, S_r \in \Omega\}. \tag{12}$$

To consider the dual contributions of a single lm-SRC, the reduced cost of a timed route can be computed as follows:

$$\hat{c}_{S_r} = \bar{c}_{S_r} - \delta_{S_r} \sigma, S_r \in \Omega, \tag{13}$$

In the above, σ is the dual variable of a lm-SRC, and $\delta_{S_r} = \alpha(N^{'}, M, k, S_r)$ is the variable coefficient of the lm-SRC. The objective of the pricing problem is then computed as follows:

$$\min \hat{c}_{S_r} = \min\{c_{S_r} - \sum_{i \in N} a_{iS_r} \pi_i - \delta_{S_r} \sigma, S_r \in \Omega\}. \tag{14}$$

4. Branch-cut-and-price (BCP) algorithm for solving the TDGVRPTW

This section presents a BCP algorithm for solving the TDGVRPTW with lm-SRCs, and the pricing algorithm used to solve the pricing problem.

4.1. Column-and-row generation

Column generation is an iterative algorithm that solves an RMP and pricing problem at each iteration; it is widely used for VRPs and their variants (Desrochers et al. 1992, Lübbecke and Desrosiers 2005, Costa et al. 2019).

At a given node of a branch-and-bound (BB) search tree, the LP-relaxation corresponds to the continuous relaxation of the set partitioning formulation, as augmented by the applicable branching decisions and lm-SRC. The LP-relaxation is solved in a column-and-row generation fashion where, at each iteration, a primal simplex algorithm is used to solve the RMP and to provide a primal and dual solution. Then, the pricing subproblem (*column generation*) is solved to find negative reduced cost columns or variables. If no negative reduced cost columns can be found, the current primal solution is optimal for the RMP. Otherwise, one or several negative reduced cost columns are added to the RMP before beginning a new iteration. If the current primal solution is optimal, the valid lm-SRC is separated in a cutting plane (*row generation*). The cutting plane algorithm terminates when no additional valid inequalities are identified by the separation algorithms, and a new iteration of the column-and-row generation is executed. The lower bound computation stops when both the column and row generation algorithms terminate without having found new columns/rows to be added to the RMP. In our implementation, the lm-SRC is separated based on the separation algorithm described by Pecin et al. (2017b).

4.2. Label-Setting Algorithm for solving the pricing problem

The pricing problem associated with the TDGVRPTW is a complex elementary shortest path problem with resource constraints (ESPPRC) and TD travel times (TDESPPRC), and can be solved efficiently using label-setting algorithms (Feillet et al. 2004, Irnich and Desaulniers 2005, Chabrier 2006, Righini and Salani 2006, 2008). Poggi and Uchoa (2014) provided an in-depth review of these topics.

In this section, we define the labels in the TDGVRPTW, and present the calculation of attributes for an arbitrary label. A new dominance rule is proposed for eliminating dominated TD arcs.

4.2.1. Label definition. In contrast to VRPs (Kelly and Xu 1999, Bräysy 2003) and GVRPs (Yu et al. 2016, Wang et al. 2018), where the vehicle speed is constant, in the TDGVRPTW, the vehicle speed changes over time. Therefore, the departure time of each node in a label needs to be determined, as different departure times may cause different carbon emissions. Here, $T(L_f)$ is used to record the departure time of each node in a label. Thus, in the forward labeling algorithm, the label $L_f = (i(L_f), q(L_f), t(L_f), S(L_f), T(L_f), c(L_f), \bar{c}(L_f), \hat{c}(L_f))$ represents a timed route that leaves node $i(L_f)$ with load $q(L_f)$, where the earliest departure time at node $i(L_f)$ is $t(L_f)$, and $S(L_f)$ is the sequence of nodes visited. The other attributes are as follows:

- $c(L_f)$: total carbon emissions along L_f;
- $\bar{c}(L_f)$: reduced cost of L_f; and
- $\hat{c}(L_f)$: reduced cost of adding the lm-SRCs of L_f.

4.2.2. Extension rules based on non-dominated TD arcs. According to Theorem 1, all labels included dominated TD arcs are unpromising. Therefore, during the extension from L_f to node $j \in N$, only non-dominated TD arcs are used.

Let $L_f^{'} = (i(L_f^{'}), q(L_f^{'}), t(L_f^{'}), S(L_f^{'}), T(L_f^{'}), c(L_f^{'}), \bar{c}(L_f^{'}), \hat{c}(L_f^{'}))$ be the expanded label. The attributes in $L_f^{'}$ are computed as follows:

$$q(L_f^{'}) = q(L_f) + q_j \tag{15}$$

$$S(L_f^{'}) = S(L_f) \cup \{j\} \tag{16}$$

$$T(L_f^{'}) = T(L_f) \cup \{t_{i(L_f)}\} \tag{17}$$

$$c(L_f^{'}) = c(L_f) + c_{i(L_f)j}(t_{i(L_f)}) \tag{18}$$

$$\bar{c}(L_f^{'}) = \bar{c}(L_f) + c_{i(L_f)j}(t_{i(L_f)}) - \pi_j \tag{19}$$

$$\hat{c}(L_f^{'}) = \bar{c}(L_f^{'}) - \delta_{S_r}\sigma. \tag{20}$$

Then, the earliest departure time $t(L_f^{'})$ of label $L_f^{'}$ is calculated as follows:

$$t(L_f^{'}) = \begin{cases} e_j + s_j & t(L_f) + \tau_{i(L_f)j}(t(L_f)) < e_j \\ t(L_f) + \tau_{i(L_f)j}(t(L_f)) + s_j & e_j \le t(L_f) + \tau_{i(L_f)j}(t(L_f)) \le l_j. \end{cases} \tag{21}$$

According to Theorem 1, the optimal timed route is composed of non-dominated TD arcs. Therefore, only non-dominated TD arcs must be used for label extension. Algorithm 1 describes the extension procedure from L_f to node $j \in N$.

Algorithm 1 evaluates the feasible extension from L_f to customer $j \in N$. In step (2), $ed_{i(L_f)j}$ and $ld_{i(L_f)j}$ are calculated, where $ed_{i(L_f)j}$ is the earliest departure time of the last

Algorithm 1 Labeling algorithm: extension from label L_f to node j

```
Input: $L_f$
Output: $\Omega_f$ (labels defined by non-dominated TD arcs)
$\Omega_f \leftarrow \emptyset$;
step(1)
for $j = 1 \rightarrow n$ do
    if $j \in S(L_f) \vee q(L_f) + q_j > Q \vee t(L_f) + \tau_{i(L_f)j}(t(L_f)) > l_j \vee A_{i(L_f)j} = \emptyset$ then
        continue
    else
        step(2)
        $ed_{i(L_f)j} \leftarrow t(L_f), ld_{i(L_f)j} \leftarrow l_j - d_{i(L_f)j}/v^{max}_{i(L_f)j}$
        step(3)
        Compute the non-dominated TD arcs from $i(L_f)$ to $j$ with $t_{i(L_f)} \in [ed_{i(L_f)j}, ld_{i(L_f)j}]$
        step(4)
        for each non-dominated TD arc do
            Compute the value of attributes in $L_f^{'}$, then add $L_f^{'}$ to $\Omega_f$.
        end for
    end if
end for
return $\Omega_f$
```

node $i(L_f)$ of L_f, and $ld_{i(L_f)j}$ is the latest departure time of $i(L_f)$. The value $v^{max}_{i(L_f)j}$ is the maximal speed in $[e_{i(L_f)}, l_j]$. The procedure used to generate the non-dominated TD arcs in step (3) of the algorithm is described in Section 2.4. After computing the departure time horizon $[ed_{i(L_f)j}, ld_{i(L_f)j}]$, the non-dominated TD arcs are obtained by dividing the departure time horizon $[ed_{i(L_f)j}, ld_{i(L_f)j}]$ into several regions and identifying their departure times in each region, and the departure times of the non-dominated TD arcs are recorded. Notably, if $[ed_{i(L_f)j}, ld_{i(L_f)j}] \cap [W_1^{'}, W_1] \neq \emptyset$ and $ed_{i(L_f)j} < W_1^{'}$, the TD arc with the earliest departure time $ed_{i(L_f)j}$ is also non-dominated. This TD arc cannot be missed. In step (4), for every non-dominated TD arc, the values of the attributes are calculated.

4.2.3. Dominance rules. According to the characteristics of the TDGVRPTW, there are many labels associated with a same node sequence, with different departure times for the depot and customers. Therefore, a new dominance rule is proposed for rapidly eliminating unpromising labels with the same node sequence. The new dominance rule is defined in Definition 6.

Definition 6. Label L_f^1 dominate label L_f^2 if

1. $S(L_f^1) = S(L_f^2)$;
2. $t(L_f^1) \leq t(L_f^2)$; and
3. $c(L_f^1) \leq c(L_f^2)$.

Condition 1 ensures that the constraints of the customers and vehicle capacity for L_f^1 and L_f^2 are the same. Condition 2 means that L_f^1 contains all of the extension possibilities of L_f^2. Condition 3 indicates that L_f^1 has lower carbon emissions than L_f^2.

To further reduce the number of labels, we use the following dominance rule.

Definition 7. Label L_f^1 dominate label L_f^2 if

1. $i(L_f^1) = i(L_f^2)$;
2. $t(L_f^1) \leq t(L_f^2)$;
3. $q(L_f^1) \leq q(L_f^2)$;
4. $\bar{S}(L_f^1) \subseteq \bar{S}(L_f^2)$; and
5. $\hat{c}(L_f^1) - \sum_{p \in P} \sigma_p \leq \hat{c}(L_f^2)$.

Condition 1 ensures that the last nodes of L_f^1 and L_f^2 are the same. Conditions 2, 3, and 4 guarantee that the feasible expansion of L_f^1 includes all of the situations of L_f^2. Owing to the addition of the lm-SRCs, Condition 5 is based on the dominance rule developed by Jepsen et al. (2008), where $P = \{p : \sigma_p < 0 \wedge \varphi_p(L_f^1) > \varphi_p(L_f^2)$ and $\varphi_p(L_f) = |N' \cap S(L_f)| \mod k$. Set $\bar{S}(L_f)$ is the set of nodes that cannot be served by the vehicle.

4.2.4. Time-dependent shortest path problem with resource constraints (TDSPPRC) as the pricing problem. Relaxing the pricing problem by allowing nonelementary paths results in a time-dependent shortest path problem with resource constraints (TDSPPRC) as a pricing problem (Irnich and Desaulniers 2005, Dabia et al. 2013). The pricing algorithm described above can be easily adapted to solve the TDSPPRC. Relaxation has the advantage of reducing the complexity in solving the pricing problem, at the price of a weaker lower bound.

4.3. Separation of the limited memory subset row cuts (lm-SRCs)

For lm-SRCs, $|N^{'}|$ and k may have different values. For example, 4-SRCs: $|N^{'}|=4$ and $k=3/2$; 5,2-SRCs: $|N^{'}|=5$ and $k=2$ (Pecin et al. 2017a,b). In this study, we adopt 3-SRCs ($|N^{'}|=3$ and $k=2$), which are the most popular lm-SRCs; these have been employed in previous studies (Baldacci et al. 2011, Jepsen et al. 2008, Pecin et al. 2017a,b). In addition, Constraints (22) correspond to a subset of the lm-SRCs.

$$\sum_{S_r\in\Omega(N^{'})} \alpha(N^{'},M,k,S_r)x_{S_r} \leq \left\lfloor \frac{|N^{'}|}{k} \right\rfloor, \forall N^{'}\in\mathcal{S} \tag{22}$$

Here, $\mathcal{S}\subseteq\{N^{'}\subset N : |N^{'}|=3\}$ is a subset of all customers' triplets; $\Omega(N^{'})\subseteq\Omega$ is the subset of the index set of all patterns containing at least two customers in $N^{'}$ (i.e., $\Omega(N^{'})=\{S_r\in\Omega : |N^{'}\cap S_r|\geq 2\}$). Let $\sigma=(\sigma_1,\sigma_2,\ldots,\sigma_{|\mathcal{S}|})$ be the dual variables associated with Constraints (22). The constraints in (22) are separated by asking whether $\sum_{S_r\in\Omega(N^{'})}\alpha(N^{'},M,k,S_r)x_{S_r}\geq\left\lfloor\frac{|N^{'}|}{k}\right\rfloor$. The inequalities are separated using the procedure described by Pecin et al. (2017b). We refer the reader to the work of Pecin et al. (2017b) for the corresponding details. If violated lm-SRCs are found, they are added to the RMP.

4.4. Branching strategy

A BB algorithm is used to obtain integer solutions (Desrosiers et al. 1984). The BB tree is explored using a best-bound strategy, and the algorithm branches on the arc variables by means of a classical branching arc strategy used for VRPs (see, for example, Costa et al. (2019)).

5. Computational results

The algorithm described in this paper was coded in C# and ILOG CPLEX 12.6 was used as the LP solver. All tests were executed on a computer with a 3.20-GHz Intel Core TM i7-8700 processor under Microsoft windows 10 operating system and 16.0 GB of RAM.

5.1. Benchmark instances and parameter settings

We used the Solomon VRPTW benchmark instances (Solomon 1987) to solve the TDGVRPTW. There are three classes of instances, divided based on the geographical distribution of the customers: R (random), C (clustered), or RC (semi-clustered).

We used a 24-hour system to modify the Solomon benchmark instances, as described in Dabia et al. (2017). The time window $[e_0,l_0]$ of the depot was $[0,24]$. Therefore, the time window $[e_i^{'}, l_i^{'}]$ of customer i in the Solomon instances was modified as $[e_i, l_i] = [e_i^{'}\frac{24}{l_0^{'}}, l_i^{'}\frac{24}{l_0^{'}}]$. The demand of customer i was $q_i = 10q_i^{'}$. The distance d_{ij} in the instances was computed as $d_{ij} = 2d_{ij}^{'}$, and the service time s_i at customer i was set as equal to $s_i = s_i^{'}\frac{24}{l_0^{'}}$. For each Solomon-based 25-customer instance, we considered only the $4^{th}, 8^{th}, \ldots, 100^{th}$ customers of the Solomon's instance with 100 customers; for each 50-customer instance, we considered only the $2^{th}, 4^{th}, \ldots, 100^{th}$ customers of the Solomon's instance with 100 customers.

The planning horizon in Figure 1 was discretized into three time periods, with different speeds associated with each period (Ichoua et al. 2003). The breakpoints and vehicle speeds were chosen as in the dataset used by Ichoua et al. (2003). The complete dataset used in this study is shown in Table 2. We adopted the parameters and typical values for the vehicles from Bektaş and Laporte (2011), Demir et al. (2012), Franceschetti et al. (2013), as shown in Table 3.

Table 2 Parameters of breakpoints and vehicle speeds.

Notation	Description	Typical values
W_1	Breakpoint 1 (h)	7
W_2	Breakpoint 2 (h)	17
W_3	Breakpoint 3 (h)	24
v_1	Vehicle speed 1 (km/h)	42
v_2	Vehicle speed 2 (km/h)	25
v_3	Vehicle speed 3 (km/h)	45

Table 3 Parameters of vehicle.

Notation	Description	Typical values
Q	Capacity (Kg)	1000
W	Weight of empty vehicle (Kg)	920
ρ	Air density (kg/m^3)	1.2041
A	Frontal surface of vehicle (m^2)	3.912
g	Gravitational constant (m/s^2)	9.81
C_d	Coefficient of aerodynamic drag	0.7
C_r	Rolling resistance	0.01
f	Fuel consumption index parameter (J/g)	1/44000
a	Acceleration (m/s^2)	0
γ	CO_2 emissions index parameter(g/g)	3.164

5.2. Computational results of the BCP for the TDGVRPTW

The computational results are given in Table 4 and Table 5 for the Solomon benchmark VRPTW instances with 25 and 50 customers, respectively.

Table 4 and Table 5 show that the BCP algorithm can efficiently solve 16 instances with 25 customers, and four instances with 50 customers. The instance (R101) with 100

Table 4 Computational results of TDGVRPTW with 25 customers.

Instance	lb_{noCut}	lb	$t_{root}(s)$	ub	$node$	$t(s)$	NV	col	cut
R101	157.67	157.67	6.25	157.67	1	6.25	6	875	0
R102	131.34	131.34	275.02	131.34	1	275.02	5	3792	0
R105	137.64	138.48	126.99	138.48	1	126.99	6	2373	5
R106	116.05	116.07	3417.01	116.07	1	3417.02	5	7097	8
R109	87.97	89.24	2819.40	89.24	1	2819.40	7	6542	21
C101	200.02	200.02	5.54	200.02	1	5.54	9	1460	0
C102	192.05	192.05	293.32	192.05	1	293.32	8	6193	0
C105	184.16	184.16	44.42	184.16	1	44.42	9	3185	0
C106	180.80	180.80	122.49	180.80	1	122.49	9	4532	0
C107	175.53	178.09	252.91	178.09	1	252.91	9	4988	6
C108	166.03	166.03	302.17	166.03	1	302.17	8	6502	0
C109	147.35	147.74	1548.43	147.74	1	1548.43	8	12255	8
RC101	164.80	164.80	40.12	164.80	1	40.12	7	2021	0
RC102	153.58	153.58	2996.26	153.58	1	2996.26	7	5744	0
RC105	144.96	145.81	766.77	145.81	1	766.77	7	3875	3
RC106	130.18	131.38	1021.80	131.84	5	3522.39	7	5034	22

[1] lb_{noCut} is the lower bound obtained in the root node without lm-SRCs;

[2] lb is the lower bound obtained in the root node;

[3] $t_{root}(s)$ is the computing time for obtaining the lb;

[4] ub is the best upper bound (i.e., the optimal solution);

[5] $node$ is the total size of the branching trees;

[6] $t(s)$ is the total running time;

[7] NV is the total number of used vehicles;

[8] col is the number of columns added to the RMP; and

[9] cut is the number of lm-SRCs added to the RMP.

Table 5 Computational results of TDGVRPTW with 50 customers.

Instance	lb_{noCut}	lb	$t_{root}(s)$	ub	$node$	$t(s)$	NV	col	cut
R101	272.24	273.18	320.32	273.18	1	320.32	13	3468	17
C101	289.13	294.17	547.84	294.17	1	547.84	11	5520	4
C105	263.81	268.66	3493.41	269.85	4	11941.58	11	11069	20
C106	263.32	267.68	5589.77	269.49	3	16466.25	11	13458	4
RC101	265.53	272.04	3357.23	-	6	30000.00	-	-	39

customers can be solved in 28194.51s, and the optimal solution is 488.49. The BCP algorithm cannot solve some instances with 25 customers, i.e., those characterized by wide time windows.

5.3. Effectiveness of the dominance rule

To demonstrate the advantages of our new dominance rule, we solved the instances with 25, 50, and 100 customers using the BCP algorithm with the new dominance rule, and the BCP without it. Table 6 presents the results of the comparison.

The results show great computational superiority when using the new dominance rule to solve the TDGVRPTW. For the instances solvable by the BCP without the new dominance

Table 6 Branch-cut-and-price (BCP) with new dominance rule vs. BCP without new dominance rule

num	Instances	Total Time with new dominance rule(s)	Total Time without new dominance rule(s)	$\triangle T(s)$	$\triangle T(\%)$
25	R101	6.25	15.55	9.30	59.81
	R102	275.02	536.51	261.49	48.74
	R105	126.99	152.94	25.95	16.97
	R106	3417.02	3526.94	109.92	3.12
	R109	2819.40	3562.44	743.04	20.86
	C101	5.54	109.09	103.55	94.92
	C102	293.32	616.47	323.15	52.42
	C105	44.42	242.58	198.16	81.69
	C106	122.49	335.70	213.21	63.51
	C107	252.91	539.04	286.13	53.08
	C108	302.17	587.72	285.55	48.59
	C109	1548.43	2242.39	693.96	30.95
	RC101	40.12	108.64	68.52	63.07
	RC102	2996.26	3296.50	300.24	9.11
	RC105	766.77	901.83	135.06	14.98
	RC106	3522.39	4256.33	733.94	17.24
50	R101	320.32	618.69	298.37	48.23
	C101	547.84	5992.74	5444.90	90.86
	C105	11941.58	56546.68	44605.10	78.88
	C106	16466.25	-	-	-
100	R101	28194.51	-	-	-

[1] num is the number of customers in the instance;

[2] $\triangle T(s)$ = computing time of the BCP without the new dominance rule - computing time of the BCP with the new dominance rule; and

[3] $\triangle T(\%) = \frac{\triangle T(s)}{\text{(computing time of the BCP without the new dominance rule)}} \times 100\%$.

rule, the BCP with the new dominance rule improves the average computational time by 47.21%. In the best case (C101 with 25 customers), the improvement is 94.92%. In instances with 50 customers (C101), the computational time is improved by up to 90.86%, and two instances (C106.50 and R101.100) unsolvable by the BCP without the new dominance rule can now be solved.

5.4. Time-dependent elementary shortest path problem with resource constraints (TDESPPRC) vs. TDSPPRC

We solved the instances with 25, 50, and 100 customers using the BCP algorithm with the new dominance rule. Table 7 presents the comparison results for the TDESPPRC and TDSPPRC.

For the instances with 25 customers, five instances (R106, R109, RC102, RC105, and RC106) cannot be solved when the pricing problem is a TDSPPRC. This is because many timed routes are generated, making even the lower bound difficult to obtain. Two instances (R102.25 and C102.25), whose pricing problem are TDSPPRCs, generated more timed

Table 7 Computational results of time-dependent elementary shortest path problem with resource constraints (TDESPPRC) and time-dependent shortest path problem with resource constraints (TDSPPRC)

		TDESPPRC						TDSPPRC					
num	instance	*lb*	*ub*	*cut*	*col*	*t*(*s*)	*node*	*lb*	*ub*	*cut*	*col*	*t*(*s*)	*node*
25	R101	157.67	157.67	0	875	6.25	1	157.67	157.67	0	875	7.23	1
	R102	131.34	131.34	0	3792	275.02	1	131.34	131.34	0	4299	805.81	1
	R105	138.48	138.48	5	2373	126.99	1	138.48	138.48	5	2373	255.57	1
	R106	116.07	116.07	8	7097	3417.02	1	-	-	-	-	7200.00	-
	R109	89.24	89.24	21	6542	2819.40	1	-	-	-	-	7200.00	-
	C101	200.02	200.02	0	1460	5.54	1	200.02	200.02	0	1460	6.51	1
	C102	192.05	192.05	0	6193	293.32	1	192.05	192.05	0	6629	392.50	1
	C105	184.16	184.16	0	3185	44.42	1	184.16	184.16	0	3185	52.47	1
	C106	180.80	180.80	0	4532	122.49	1	180.80	180.80	0	4532	133.73	1
	C107	178.09	178.09	6	4988	252.91	1	178.09	178.09	6	4988	263.94	1
	C108	166.03	166.03	0	6502	302.17	1	166.03	166.03	0	6502	288.23	1
	C109	147.74	147.74	8	12255	1548.43	1	147.74	147.74	8	12255	1805.31	1
	RC101	164.80	164.80	0	2021	40.12	1	164.80	164.80	0	2021	45.22	1
	RC102	153.58	153.58	0	5744	2996.26	1	-	-	-	-	7200.00	-
	RC105	145.81	145.81	3	3875	766.77	1	-	-	-	-	7200.00	-
	RC106	131.38	131.84	22	5034	3522.39	5	-	-	-	-	7200.00	-
50	R101	273.18	273.18	17	3468	320.32	1	273.18	273.18	17	3468	300.18	1
	C101	294.17	294.17	4	5520	547.84	1	294.17	294.17	4	5520	556.00	1
	C105	268.66	269.85	20	11069	11941.58	4	268.66	269.85	20	11069	12407.13	4
	C106	267.68	269.49	4	13458	16466.25	3	267.68	269.49	4	13458	17320.25	3
100	R101	488.49	488.49	6	13448	28194.51	1	488.49	488.49	6	13448	28176.93	1

routes than the instances whose pricing problems were TDESPPRCs. We observe that the TDSPPRC results in good performance for three instances (C108.25, R101.50 and R101.100). However, the effect is not significant.

5.5. Effectiveness of the lm-SRCs

As shown in Table 8, 12 instances were solved based on using BCP with lm-SRCs and BP without lm-SRCs approaches, respectively.

Table 8 Branch-cut-and-price algorithm (BCP) vs. Branch-and-price (BP) algorithm

		BCP						BP					
num	instance	*lb*	*ub*	*cut*	*col*	*t*(*s*)	*node*	*lb*	*ub*	t_{root}	*col*	*t*(*s*)	*node*
25	R105	138.48	138.48	5	2373	126.99	1	137.64	138.48	86.69	2211	337.77	4
	R106	116.07	116.07	8	7097	3417.02	1	116.05	116.07	2287.53	7095	4932.16	4
	R109	89.24	89.24	21	6542	2819.40	1	87.97	89.24	1602.76	6539	10438.97	10
	C107	178.09	178.09	6	4988	252.91	1	175.53	178.09	179.14	4984	519.95	3
	C109	147.74	147.74	8	12255	1548.43	1	147.35	147.74	944.84	12173	3697.40	5
	RC105	145.81	145.81	3	3875	766.77	1	144.96	145.81	722.67	3875	3016.96	9
	RC106	131.38	131.84	22	5034	3522.39	5	130.18	131.84	724.89	5034	4652.60	13
50	R101	273.18	273.18	17	3468	320.32	1	272.24	273.18	219.59	3457	2140.69	13
	C101	294.17	294.17	4	5520	547.84	1	289.13	294.17	365.35	5487	1129.37	4
	C105	268.66	269.85	20	11069	11941.58	4	263.81	269.85	1743.35	10949	12717.31	7
	C106	267.68	269.49	4	13458	16466.25	3	263.32	269.49	2726.12	13138	31396.39	9
100	R101	488.49	488.49	6	13448	28194.51	1	488.34	488.49	20898.69	13443	42012.89	3

For these 12 instances, the BCP approach performs better. The computational times are improved by up to 62.40%, 30.72%, 72.99%, 51.36%, 58.12%, 74.58%, 24.29%, 85.04%, 51.49%, 6.10%, 47.55%, and 32.89%, respectively. And the BCP algorithm improves the average computational time by 49.80%.

5.6. Carbon emissions vs. waiting time

In the TDGVRPTW, we define two types of waiting times (see Section 2.2). As waiting at each node (depot and customers) may lead to reduced carbon emissions, we compared the effect of the Type 2 waiting time on carbon emissions.

According to the computational results from Subsection 5.2, we removed all of the Type 2 waiting times of each node (depot and customers) without changing the sequence of nodes, and recalculated the carbon emissions for each timed route. For each computational result, the two types of waiting times and the gap in the carbon emissions are shown in Table 9. As shown, waiting can have a positive effect on the reduction of carbon emissions.

Table 9 Carbon emissions vs. Waiting time

		Solution of TDGVRPTW						Solution without type2 waiting				
num	instance	ce_w	t_ret_w	$type1\_wt$	$type2\_wt$	$rat\_1\_w(\%)$	$rat\_2(\%)$	ce_n	t_ret_n	$type1\_wt$	$rat\_1\_n(\%)$	$rat\_ce(\%)$
25	R101	157.67	108.30	34.14	20.67	31.52	19.09	173.94	107.44	52.24	48.62	9.36
	R102	131.34	99.18	24.98	23.47	25.19	23.66	147.26	99.18	47.96	48.36	10.81
	R105	138.48	103.96	25.38	36.66	24.41	35.26	158.91	102.63	47.48	46.26	12.86
	R106	116.07	90.79	12.15	36.47	13.38	40.17	135.46	88.81	35.33	39.78	14.31
	R109	89.24	117.82	10.26	87.28	8.71	74.08	134.23	116.43	57.47	49.36	33.52
	C101	200.02	157.13	35.60	34.03	22.66	21.66	230.92	157.13	70.79	45.05	13.38
	C102	192.05	138.99	28.61	25.60	20.58	18.42	216.89	138.99	54.06	38.89	11.45
	C105	184.16	155.40	29.63	41.94	19.07	26.99	221.88	154.56	67.79	43.86	17.00
	C106	180.80	152.33	28.65	47.08	18.81	30.91	217.67	150.11	65.03	43.32	16.94
	C107	178.09	151.41	22.97	48.95	15.17	32.33	221.88	149.77	63.08	42.12	19.74
	C108	166.03	135.06	11.94	54.83	8.84	40.60	208.62	129.98	46.29	35.61	20.40
	C109	147.74	135.99	2.61	98.75	1.92	72.62	201.97	124.31	39.92	32.11	26.85
	RC101	164.80	125.23	22.72	56.36	18.14	45.01	206.22	123.16	59.92	48.65	20.08
	RC102	153.58	125.65	20.55	72.09	16.35	57.37	197.41	123.60	61.36	49.64	22.20
	RC105	145.81	122.55	21.09	54.23	17.21	44.25	191.27	121.98	56.86	46.61	23.77
	RC106	131.84	119.50	5.55	96.44	4.64	80.70	196.85	109.47	45.42	41.49	33.02
50	R101	273.18	235.71	77.93	47.49	33.06	20.15	305.87	235.71	124.76	52.93	10.69
	C101	294.17	196.54	34.37	24.15	17.49	12.29	321.43	196.54	58.04	29.53	8.48
	C105	269.85	191.22	22.03	40.51	11.52	21.19	301.76	190.20	51.84	27.26	10.57
	C106	269.49	192.23	22.36	48.02	11.63	24.98	302.84	186.87	48.64	26.03	11.01
100	R101	488.49	345.14	92.17	55.54	26.71	16.09	525.54	345.14	141.04	40.86	7.05

[1] ce_w is the total carbon emissions with the type 2 waiting time in each node in the instance;

[2] t_ret_w is the total return time to depot $n+1$ with the type 2 waiting time in each node;

[3] $rat\_1\_w(\%) = type1\_wt/t\_ret\_w \times 100\%$;

[4] $rat\_2(\%) = type2\_wt/t\_ret\_w \times 100\%$;

[5] ce_n is the total carbon emissions without the type 2 waiting time in each node;

[6] t_ret_n is the total return time to depot $n+1$ without the type 2 waiting time in each node;

[7] $rat\_1\_n(\%) = type1\_wt/t\_ret\_n \times 100\%$; and

[8] $rat\_ce(\%) = ce\_n - ce\_w/ce\_n \times 100\%$.

For instance, in C101.25, the Type 1 waiting time is reduced by 35.19 and the Type 2 waiting time increases to 34.03, but the carbon emissions are reduced 30.90 (13.38%). Some of the Type 1 waiting times (35.19) are converted into Type 2 waiting times (34.03). This means that the total waiting time when considering carbon emissions is slightly decreased (69.63 vs. 70.79), and that the carbon emissions can be reduced by rebalancing the waiting times of the TDGVRPTW.

5.7. Sensitivity analyses on speed profiles

In this subsection, we consider two speed profiles to analyze how the carbon emissions change based on different speed profiles. In addition to the profile considered in Section 2.1 (see Figure 1), we added a similar profile, but with a congestion speed (i.e., v_2) equal to 30 km/h, similar to (Jabali et al. 2012), instead of 25 km/h. These two speed profiles were randomly assigned to various arcs in the instances. Table 10 presents the computation results using the two speed profiles.

Table 10 One speed profile vs. Two speed profiles

		One speed profile			Two speed profiles			
num	instance	*ce_1*	*type1_wt*	*type2_wt*	*ce_2*	*type1_wt*	*type2_wt*	*rat_ce*(%)
25	R101	157.67	34.14	20.67	161.41	37.37	21.00	2.37
	R102	131.34	24.98	23.47	135.80	19.85	23.67	3.40
	R105	138.48	25.38	36.66	143.16	26.45	33.91	3.38
	R106	116.07	12.15	36.47	120.17	15.23	34.45	3.53
	R109	89.24	10.26	87.28	98.72	6.32	80.95	10.62
	C101	200.02	35.60	34.03	212.86	39.21	33.47	6.42
	C102	192.05	28.61	25.60	197.46	29.17	26.59	2.82
	C105	184.16	29.63	41.94	200.38	27.93	42.94	8.81
	C106	180.80	28.65	47.08	188.34	29.29	46.72	4.17
	C107	178.09	22.97	48.95	186.87	21.03	33.42	4.93
	C108	166.03	11.94	54.83	172.52	10.84	32.59	3.91
	C109	147.74	2.61	98.75	158.50	3.96	65.28	7.28
	RC101	164.80	22.72	56.36	171.73	18.29	67.76	4.21
	RC102	153.58	20.55	72.09	157.45	21.06	72.05	2.52
	RC105	145.81	21.09	54.23	156.20	19.48	61.93	7.13
	RC106	131.84	5.55	96.44	137.29	2.22	92.81	4.13
50	R101	273.18	77.93	47.49	283.76	72.79	40.31	3.87
	C101	294.17	34.37	24.15	304.35	35.61	24.83	3.46
	C105	269.85	22.03	40.51	284.48	22.64	42.20	5.42
	C106	269.49	22.36	48.02	280.47	25.68	44.87	4.07

[1] *ce_1* is the total carbon emissions with one speed profile in an instance;

[2] *ce_2* is the total carbon emissions with two speed profiles; and

[3] $rat\_ce(\%) = ce\_2 - ce\_1/ce\_1 \times 100\%$.

From the computational results, it can be seen that for the two speed profile cases, the carbon emissions increase by an average of 4.82%.

5.8. Solving the TDVRPTW to minimize the total travel time

In addition to solving the TDGVRPTW, our BCP algorithm can also be used to solve a wide class of TDVRPTWs, with an objective function related to the departure times from the depot and customer locations (e.g., the total travel time). In this subsection, we solve the instance C101.25 of the TDVRPTW to minimize the total travel time, and we compute the corresponding carbon emissions from the resulting optimal solutions. Subsequently, we solve the instance C101.25 of the TDGVRPTW to minimize the total carbon emissions, and we compute the corresponding travel time for the resulting optimal solutions.

Table 11 Solution of C101.25

Objective	Route ID	Route with departure time (timed route)	tt	ce	ttt	tce
TDVRPTW to minimize the travel time	1	0[0.00]-4[10.97]-3[17.01]-$n+1$	4.01	37.32	34.14	317.75
	2	0[0.00]-8[3.25]-9[17.01]-$n+1$	3.41	34.35		
	3	0[0.00]-14[9.22]-15[17.01]-$n+1$	4.56	42.88		
	4	0[0.00]-19[5.69]-20[17.01]-$n+1$	5.31	52.88		
	5	0[0.00]-21[10.64]-22[17.01]-$n+1$	3.11	27.74		
	6	0[0.00]-5[2.22]-6[4.21]-7[17.01]-$n+1$	1.76	17.65		
	7	0[0.00]-18[10.49]-16[14.02]-17[17.01]-$n+1$	2.69	21.84		
	8	0[0.00]-24[3.59]-23[8.89]-25[17.01]-$n+1$	4.78	40.30		
	9	0[0.00]-10[6.87]-11[8.78]-12[17.01]-13[19.46]-$n+1$	2.66	24.23		
	10	0[0.00]-2[6.70]-1[17.01]-$n+1$	1.86	18.55		
TDGVRPTW to minimize the carbon emissions	1	0[5.73]-14[10.21]-15[12.66]-$n+1$	6.76	22.91	46.36	200.02
	2	0[0.00]-19[7.01]-20[16.68]-$n+1$	5.81	48.53		
	3	0[7.01]-21[11.24]-22[14.27]-$n+1$	5.07	10.42		
	4	0[0.00]-10[7.01]-11[9.00]-12[14.02]-$n+1$	3.41	13.89		
	5	0[0.00]-24[7.01]-23[9.45]-25[14.31]-$n+1$	6.25	26.82		
	6	0[0.00]-2[7.01]-4[10.97]-3[14.41]-$n+1$	6.09	21.17		
	7	0[0.00]-5[2.22]-6[7.01]-7[12.35]-1[15.94]-$n+1$	4.26	15.85		
	8	0[0.00]-8[7.01]-9[14.66]-13[19.46]-$n+1$	4.80	29.15		
	9	0[7.01]-18[10.67]-16[14.02]-17[16.01]-$n+1$	3.91	11.28		

[1] *tt* is the travel time;

[2] *ce* is the carbon emissions;

[3] *ttt* is the total travel time;

[4] *tce* is the total carbon emissions;

[5] For a route with departure times (timed route), the numbers in brackets indicate the departure times of the nodes. For example of timed route 1 ($0[0.00]-4[10.97]-3[17.01]-n+1$), the node sequence is $0-4-3-n+1$, the departure times are 0.00, 10.97, and 17.01 of node 0, node 4, and node 3 respectively. Depot $n+1$ does not have departure time.

We observe that the optimal solution of TDVRPTW is not the optimal solution of TDGVRPTW and that the difference in term of carbon emissions is significant. Indeed, the percentage gaps between the two solutions in term of carbon emissions is 37.05%. The percentage gaps between the two solutions in term of travel time is 26.36%. All the benchmark instances used in this paper and the results are available for download at https://github.com/liuyimingNEU95/Supplement_BCP_TDGVRPTW.git

6. Conclusions

In this paper, we described a BCP algorithm for solving the TDGVRPTW, aiming to reduce the total carbon emissions. Because a later departure time from a node may lead to less carbon emissions, the departure time of each node must be determined. We introduced the concepts of optimal timed routes and non-dominated TD arcs in the TDGVRPTW to reduce the solution space, so that timed routes with dominated TD arcs are eliminated.

Our extensive computational studies show that our BCP algorithm can solve Solomon-based instances with up to 100 customers. In addition, the results show that the new dominance rule used in the BCP can reduce the average computation time by approximately 47.21%.

Apart from solving the TDGVRPTW, future works will address other variants of the TDGVRPTW that have been proposed to consider different objectives arising from practical applications, such as the TD green pick-up and delivery VRP, TD green multi-trip VRP, and TD green multi-depot VRP.

Acknowledgments

The authors would like to thank the anonymous reviewers and associate editor for their extremely helpful suggestions and very thorough review of the paper. This research is supported by the National Natural Science Foundation of China (71831003, 71571037, 71831006, 71420107028, 71601089, 71620107003, and 71901180), and the Fundamental Research Funds for the Central Universities (N170405005, N180704015).

References

Andelmin J, Bartolini E (2017) An exact algorithm for the green vehicle routing problem. *Transportation Science* 51(4):1288–1303.

Baldacci R, Hadjiconstantinou E, Mingozzi A (2004) An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Operations Research* 52(5):723–738.

Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research* 59(5):1269–1283.

Baldacci R, Mingozzi A, Roberti R (2012) Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research* 218(1):1–6.

Bektaş T, Demir E, Laporte G (2016) Green vehicle routing. *Green Transportation Logistics*, 243–265 (Springer).

Bektaş T, Laporte G (2011) The pollution-routing problem. *Transportation Research Part B: Methodological* 45(8):1232–1250.

Boland NL, Savelsbergh MW (2019) Perspectives on integer programming for time-dependent models. *TOP* 27(2):147–173.

Bräysy O (2003) A reactive variable neighborhood search for the vehicle-routing problem with time windows. *INFORMS Journal on Computing* 15(4):347–368.

Chabrier A (2006) Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research* 33(10):2972–2990.

Çimen M, Soysal M (2017) Time-dependent green vehicle routing problem with stochastic vehicle speeds: An approximate dynamic programming algorithm. *Transportation Research Part D: Transport and Environment* 54:82–98.

Costa L, Contardo C, Desaulniers G (2019) Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science* 53(4):946–985.

Dabia S, Demir E, Van Woensel T (2017) An exact approach for a variant of the pollution-routing problem. *Transportation Science* 51(2):607–628.

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

Demir E, Bektaş T, Laporte G (2012) An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research* 223(2):346–359.

Demir E, Bektaş T, Laporte G (2014a) The bi-objective pollution-routing problem. *European Journal of Operational Research* 232(3):464–478.

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

Desaulniers G, Madsen OB, Ropke S (2014) Chapter 5: The vehicle routing problem with time windows. *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, 119–159 (SIAM).

Desrochers M, Desrosiers J, Solomon M (1992) A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 40(2):342–354.

Desrosiers J, Soumis F, Desrochers M (1984) Routing with time windows by column generation. *Networks* 14(4):545–565.

Donati AV, Montemanni R, Casagrande N, Rizzoli AE, Gambardella LM (2008) Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research* 185(3):1174–1191.

Feillet D, Dejax P, Gendreau M, Gueguen C (2004) An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal* 44(3):216–229.

Franceschetti A, Demir E, Honhon D, Van Woensel T, Laporte G, Stobbe M (2017) A metaheuristic for the time-dependent pollution-routing problem. *European Journal of Operational Research* 259(3):972–991.

Franceschetti A, Honhon D, Van Woensel T, Bektaş T, Laporte G (2013) The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological* 56:265–293.

Fukasawa R, He Q, Song Y (2016) A branch-cut-and-price algorithm for the energy minimization vehicle routing problem. *Transportation Science* 50(1):23–34.

Hashimoto H, Yagiura M, Ibaraki T (2008) An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization* 5(2):434–456.

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

Irnich S, Desaulniers G (2005) Shortest path problems with resource constraints. *Column Generation*, 33–65 (Springer).

Jabali O, Van Woensel T, De Kok A (2012) Analysis of travel times and CO_2 emissions in time-dependent vehicle routing. *Production and Operations Management* 21(6):1060–1074.

Jepsen M, Petersen B, Spoorendonk S, Pisinger D (2008) Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research* 56(2):497–511.

Kazemian I, Rabbani M, Farrokhi-Asl H (2018) A way to optimally solve a green time-dependent vehicle routing problem with time windows. *Computational and Applied Mathematics* 37(3):2766–2783.

Kelly JP, Xu J (1999) A set-partitioning-based heuristic for the vehicle routing problem. *INFORMS Journal on Computing* 11(2):161–172.

Kuo Y (2010) Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering* 59(1):157–165.

Laporte G (2009) Fifty years of vehicle routing. *Transportation Science* 43(4):408–416.

Lin C, Choy KL, Ho GT, Chung SH, Lam H (2014) Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications* 41(4):1118–1138.

Lübbecke ME, Desrosiers J (2005) Selected topics in column generation. *Operations Research* 53(6):1007–1023.

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

Malandraki C, Dial RB (1996) A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operational Research* 90(1):45–55.

Moghdani R, Salimifard K, Demir E, Benyettou A (2021) The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production* 279:123691.

Pan B, Zhang Z, Lim A (2021) Multi-trip time-dependent vehicle routing problem with time windows. *European Journal of Operational Research* 291(1):218–231.

Pecin D, Contardo C, Desaulniers G, Uchoa E (2017a) New enhancements for the exact solution of the vehicle routing problem with time windows. *INFORMS Journal on Computing* 29(3):489–502.

Pecin D, Pessoa A, Poggi M, Uchoa E (2017b) Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation* 9(1):61–100.

Poggi M, Uchoa E (2014) Chapter 3: New exact algorithms for the capacitated vehicle routing problem. *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, 59–86 (SIAM).

Righini G, Salani M (2006) Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization* 3(3):255–273.

Righini G, Salani M (2008) New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks: An International Journal* 51(3):155–170.

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

Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35(2):254–265.

Sun W, Yu Y, Wang J (2019) Heterogeneous vehicle pickup and delivery problems: Formulation and exact solution. *Transportation Research Part E: Logistics and Transportation Review* 125:181–202.

Toth P, Vigo D (2014) *Vehicle routing: Problems, methods, and applications* (SIAM).

USEPA (2021) Inventory of u.s. greenhouse gas emissions and sinks. Website, `https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks`.

Vidal T, Crainic TG, Gendreau M, Prins C (2014) A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* 234(3):658–673.

Vu DM, Hewitt M, Boland N, Savelsbergh M (2020) Dynamic discretization discovery for solving the time-dependent traveling salesman problem with time windows. *Transportation Science* 54(3):703–720.

Wang J, Yu Y, Tang J (2018) Compensation and profit distribution for cooperative green pickup and delivery problem. *Transportation Research Part B: Methodological* 113:54–69.

Yu Y, Tang J, Li J, Sun W, Wang J (2016) Reducing carbon emission of pickup and delivery using integrated scheduling. *Transportation Research Part D: Transport and Environment* 47:237–250.

Yu Y, Wang S, Wang J, Huang M (2019a) A branch-and-price algorithm for the heterogeneous fleet green vehicle routing problem with time windows. *Transportation Research Part B: Methodological* 122:511–527.

Yu Y, Wu Y, Wang J (2019b) Bi-objective green ride-sharing problem: Model and exact method. *International Journal of Production Economics* 208:472–482.