The pickup and delivery problem with time windows and handling operations

Veenstra, Marjolein; Cherkesly, Marilene; Desaulniers, Guy; Laporte, Gilbert

Published in:
Computers & Operations Research

DOI:
10.1016/j.cor.2016.07.014

Publication date:
2017

Computers & Operations Research 77 (2017) 127–140

Contents lists available at ScienceDirect

Computers & Operations Research

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

The pickup and delivery problem with time windows and handling operations

Marjolein Veenstra [a,*], Marilène Cherkesly [b], Guy Desaulniers [c], Gilbert Laporte [d]

[a] University of Groningen, P.O. Box 800, 9700 AV Groningen, The Netherlands
[b] Université du Québec à Montréal and GERAD, C.P. 8888, succursale Centre-ville, Montréal, Canada H3C 3P8
[c] École Polytechnique de Montréal and GERAD, C.P. 6079, succursale Centre-ville, Montréal, Canada H3C 3A7
[d] HEC Montréal and CIRRELT, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

ARTICLE INFO

Article history:
Received 15 January 2016
Received in revised form
5 July 2016
Accepted 20 July 2016
Available online 30 July 2016

Keywords:
Vehicle routing with pickups and deliveries
Handling operations
Column generation
Branch-price-and-cut
Valid inequalities

ABSTRACT

This paper introduces the pickup and delivery problem with time windows and handling operations. In this problem, the loading compartment of a vehicle is modeled as a linear LIFO stack. When an item is picked up, it is positioned on top of the stack. When it is on top of the stack, it can be delivered without additional handling. Otherwise, items on top must be unloaded before the delivery and reloaded afterwards, which requires time. We define two rehandling policies. For both policies, rehandling is only allowed at delivery locations and there is no specific reloading order for the rehandled items. Under the first policy, only compulsory rehandling is allowed. Under the second policy, in addition to compulsory rehandling, preventive rehandling is allowed. For each policy, we propose a branch-price-and-cut algorithm with an ad hoc dominance criterion for the labeling algorithm used to generate routes. Computational results are reported on benchmark instances for the pickup and delivery problem with time windows.

1. Introduction

In pickup and delivery problems, a fleet of vehicles based at a depot is used to complete a set of requests. A request consists of transporting an item (which can consist of multiple units) from a specific location, where the item is loaded, to a specific location, where it is unloaded. A time window is given for each pickup or delivery location, specifying the time interval during which service must start. We consider a fleet of homogeneous vehicles of limited capacity, where the compartment is rear-loaded and operated in a last-in-first-out (LIFO) fashion. The compartment is modeled as a linear LIFO stack. This implies that when an item is picked up, it is positioned on top of the stack. Therefore, an item is accessible for delivery if it is on top of the stack. Otherwise, the items on top must be unloaded before the delivery of the item and reloaded afterwards, which requires supplementary time. We define a rehandling operation as the unloading and reloading operations of an item at a pickup or delivery location. A handling operation can refer to a rehandling operation, loading an item at its pickup location, or unloading an item at its delivery location. We indicate this problem in the remainder as the pickup and delivery problem with time windows and handling operations (PDPTWH). Let 0, i^+ and i^- denote the depot, and the pickup and delivery locations corresponding to request i, respectively. Fig. 1 illustrates two routes, where route (a) does not require rehandling, whereas in route (b) item 2 needs to be rehandled before delivering item 1.

We introduce and analyze two different rehandling policies. Because an item needs to be delivered at a delivery location, the customer will allow items to be rehandled if its item is not on top of the stack. On the other hand, at a pickup location, because other vehicles from different suppliers may wait to load or unload, the customer might not allow items to be rehandled. Therefore, rehandling operations are only allowed at delivery locations for both policies. We assume that it is not possible to stop at a random location in the route to do the rehandling, which implies that eventual rehandling operations begin at the same time as the service. Therefore, rehandling operations must start within the time window of the delivery location where rehandling occurs. For both policies, there is no specific reloading order for the rehandled items. We define two items i and j to be at the same level if the most recent handling operation for both items occurred at the same location. Item i is said to be on top of item j if the most recent handling operation for item i occurred after the most recent

* Corresponding author.
E-mail addresses: marjolein.veenstra@rug.nl (M. Veenstra), marilene.cherkesly@gerad.ca (M. Cherkesly), guy.desaulniers@gerad.ca (G. Desaulniers), gilbert.laporte@cirrelt.ca (G. Laporte).

http://dx.doi.org/10.1016/j.cor.2016.07.014
0305-0548/

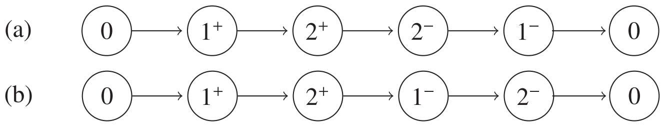

Fig. 1. Route (a) does not require rehandling and route (b) requires one rehandling operation before delivering item 1.

handling operation for item *j*. Under the first rehandling policy, called policy 1, only compulsory rehandling is allowed, i.e., all and only the items on top of the delivered item must be rehandled. Note that policy 1 forbids the rehandling of items that are on the same level as, or below, the delivered item. The second rehandling policy, called policy 2, is a generalization of policy 1. Under policy 2, compulsory rehandling must be done and preventive rehandling is allowed, i.e., all items can be rehandled at once. Fig. 2 depicts a route and its corresponding vehicle configuration under policy 1, while Fig. 3 depicts the same route and an example of a corresponding vehicle configuration under policy 2. In Fig. 2, two items are rehandled upon delivering item 1, namely items 2 and 3. Item 4 is rehandled upon delivering item 2 and again upon delivering item 3. In Fig. 3, two items are rehandled upon delivering item 1, namely items 2 and 3 and two items are rehandled upon delivering item 2, namely items 3 and 4. Since items 2 and 3 are on the same level when delivering item 2, rehandling item 3 is done preventively. Since preventive rehandling operations are not allowed under policy 1, the vehicle configuration in Fig. 3 is infeasible under policy 1. Because each rehandling operation requires additional time, it may happen that the time windows are respected under policy 2, but not under policy 1. A vehicle route is feasible if (i) the capacity of the vehicle is always respected, (ii) the time windows are respected, (iii) the pickup location of a request is visited before its corresponding delivery location, and (iv) the rehandling policy is respected. We denote by PDPTWH-1 and PDPTWH-2 the PDPTWH under policies 1 and 2, respectively. The goal of the PDPTWH is to compute feasible routes that first minimize the number of vehicles and then the total travel costs.

The PDPTWH arises in the transportation of heavy, dangerous or large items in a less-than-truckload setting. To our knowledge, the PDPTWH has not previously been studied, but several variants of this problem have been investigated, such as the pickup and delivery problem (see Berbeglia et al. [2], Parragh et al. [14,15], and Savelsbergh and Sol [18], for surveys), the pickup and delivery problem with time windows (PDPTW), the pickup and delivery problem with time windows and LIFO loading (PDPTWL), which prohibits rehandling operations, the traveling salesman problem with pickups and deliveries and handling costs (TSPPD-H), where two types of items are considered, those transported from the depot to customers and those transported from customers to the depot, and the pickup and delivery traveling salesman problem with handling costs (PDTSPH), where only compulsory rehandling is allowed and the reloading sequence is given.

Ropke et al. [17] proposed a branch-price-and-cut algorithm for the PDPTW that solves instances with up to 96 requests to optimality within two hours, on a computer equipped with an AMD Opteron 250 processor (2.4 GHz). State-of-the-art algorithms for the PDPTWL were developed by Cherkesly et al. [5,6]. Cherkesly et al. [5] proposed branch-price-and-cut algorithms that can solve instances with up to 75 requests to optimality within one hour, on a computer equipped with an Intel Core i7-3770 processor (3.4 GHz), while Cherkesly et al. [6] developed a population-based metaheuristic that solves instances with up to 300 requests within three hours, on a computer equipped with an Intel (R) Xeon(R) X5675 processor (3.07 GHz). For the instances with known optimal values, the average optimality gap obtained with their algorithm ranges between 0.17% and 0.34%. Battarra et al. [1] proposed two exact algorithms to solve the TSPPD-H under different handling policies: the first one is a branch-and-cut approach, while the second one combines Benders decomposition and branch-and-cut. The tests were run on a computer equipped with an AMD Athlon 64 $\times$ 2 Dual processor (2.20 GHz), and instances with up to 25 customers were solved within two hours. Erdoǧan et al. [10] developed heuristics for the TSPPD-H that can solve instances with up to 200 customers. The experiments were performed on a computer equipped with an Intel Core 2 Quad processor (2.83 GHz). The combination of tabu search and exact dynamic programming performs best, resulting in an average percentage deviation of 0.07% from the best known solutions. The largest instances were solved in approximately one hour on average. Veenstra et al. [19] proposed a heuristic for the PDTSPH, but the authors did not report optimality gaps for the larger instances.

This work is rooted in two different streams of research, namely, pickup and delivery routing with LIFO loading, and pickup and delivery routing with handling operations. Ropke and Cordeau [16] developed a branch-price-and-cut algorithm for the PDPTW, in which several families of valid inequalities are introduced. Cherkesly et al. [5] developed three branch-price-and-cut algorithms for the PDPTWL. They proposed an ad hoc dominance criterion and a labeling algorithm for the elementary shortest path problem with pickups and deliveries, time windows, capacity, and LIFO constraints. Cherkesly et al. [4] introduced the pickup and delivery problem with multiple stacks (PDPTWMS) and implemented two branch-price-and-cut algorithms. They adapted the hybrid branch-price-and-cut algorithm of Cherkesly et al. [5] for the PDPTWL to the PDPTWMS. Battarra et al. [1] proposed three handling policies for the TSPPD-H. Under the first policy, all items delivered at the depot are positioned on top of the items delivered at the customers, whereas under the second policy all items delivered at customers are positioned on top of the items delivered at the depot. The third policy is a hybrid between the first two. We extend these ideas to develop rehandling policies for our problem where items are transported from specific pickup locations to specific delivery locations. We propose branch-price-and-cut algorithms based on those of Ropke and Cordeau [16] and Cherkesly et al. [4,5].

The goal of this paper is to model the PDPTWH and to develop

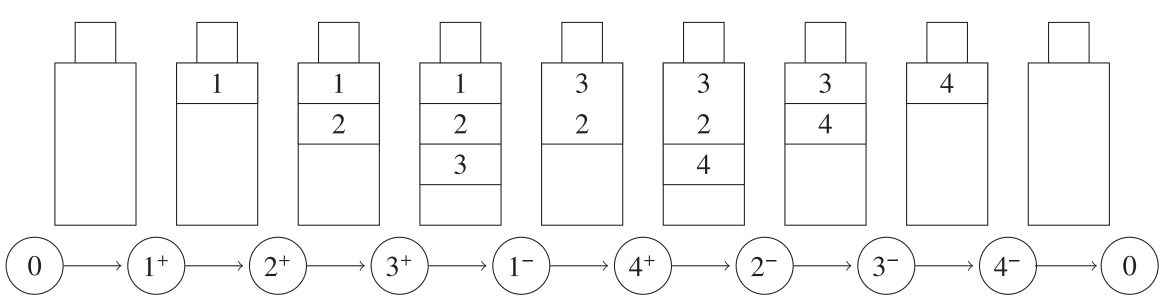

Fig. 2. Example of a route with its corresponding vehicle configuration under policy 1. There is no separation line between items at the same level.

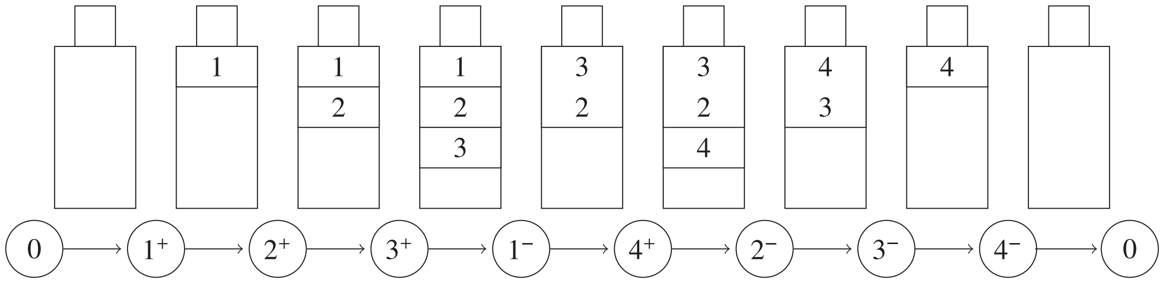

Fig. 3. Example of a route with a corresponding vehicle configuration under policy 2.

for each rehandling policy a specific branch-price-and-cut algorithm, with an ad hoc dominance criterion for the labeling algorithm used to generate routes. The structure of the paper is the following. Section 2 proposes a set partitioning formulation for the PDPTWH. Section 3 introduces branch-price-and-cut algorithms for the PDPTWH-1 and the PDPTWH-2. Computational results are reported in Section 4, followed by conclusions in Section 5.

2. Problem statement and mathematical formulation

Let n be the number of requests. The PDPTWH is defined on the graph $G = (N, A)$, where N is the set of nodes and A is the set of arcs. Let $N = \{0, 1, \dots, 2n, 2n + 1\}$, where 0 and $2n + 1$ correspond to the origin and destination depot, respectively, $P = \{1, \dots, n\}$ corresponds to the set of pickup nodes, and $D = \{n + 1, \dots, 2n\}$ corresponds to the set of delivery nodes. Request i is associated with a pickup node $i \in P$ and a delivery node $n + i \in D$, denoted as i^+ and i^-. For ease of notation, we also denote the set of requests by P. A service time s_i is defined for each node $i \in N$, where $s_i > 0$, $\forall\, i \in P \cup D$, and $s_i = 0$, $\forall\, i \in \{0, 2n + 1\}$. A time window $[\underline{w}_i, \overline{w}_i]$ is associated with each node $i \in N$, where $\underline{w}_i$ and $\overline{w}_i$ represent the earliest and the latest time at which the service and rehandling operations can begin, respectively. The time windows at the depot are assumed to be unconstraining. A homogeneous fleet of vehicles with capacity Q is given. Each node is associated with a load to be picked up or delivered, that is, $q_i > 0$, $\forall\, i \in P$, $q_i = -q_{i-n}$, $\forall\, i \in D$, and $q_i = 0$, $\forall\, i \in N \setminus \{P \cup D\}$. The time to rehandle one unit of an item is a constant value δ, thus the total time to rehandle the item $i \in P$ is δq_i. With each arc $(i, j) \in A$ are associated a travel time t_{ij} and a travel cost c_{ij}. Note that the cost c_{0i} on each arc $(0, i)$, $i \in P$, may include a vehicle fixed cost sufficiently large to ensure the minimization of the number of vehicles used.

A route is feasible if the pickup and delivery constraints are satisfied, i.e., if for every pickup node $i \in P$ visited, the corresponding delivery node $n + i$ is visited afterwards, the capacity is respected, and the time windows are respected. Given a route $r = \{i_0, i_1, \dots, i_\rho\}$ with $i_0 = 0$ and $i_\rho = 2n + 1$ that satisfies the pickup and delivery constraints, the load l_{i_k} at each visited node i_k, $k \in \{0, 1, \dots, \rho\}$, can be computed recursively as follows:

$$l_{i_0} = 0 \tag{1}$$

$$l_{i_k} = l_{i_{k-1}} + q_{i_k}, \quad \forall\; k = 1, \dots, \rho. \tag{2}$$

Route r respects the vehicle capacity if $l_{i_k} \le Q$ for all $k = 1, \dots, \rho$. Furthermore, if O_{i_k} represents the subset of requests whose items are rehandled at node i_k, $k \in \{0, 1, 2, \dots, \rho\}$ then the start of service time t_{i_k} at each visited node i_k, $k \in \{0, 1, \dots, \rho\}$, can be computed recursively as follows:

$$t_{i_0} = 0 \tag{3}$$

$$t_{i_k} = \max\left\{ \underline{w}_{i_k},\, t_{i_{k-1}} + s_{i_{k-1}} + \sum_{i \in O_{i_{k-1}}} \delta q_i + t_{i_{k-1}, i_k} \right\}. \tag{4}$$

Route r respects the time windows if $t_{i_k} \le \overline{w}_{i_k}$ for all $k = 1, \dots, \rho$.

The PDPTWH consists of finding feasible routes such that each request is completed and the total cost is minimized. We propose a set partitioning formulation for the PDPTWH, in which the rehandling operations can occur both at the pickup and delivery locations, as long as the rehandling operations start at the same time as the service, thus within the time windows. Let Ω denote the set of all feasible routes. We define c_r as the cost of route $r \in \Omega$. Let a_{ir} be a binary constant equal to one if and only if route $r \in \Omega$ completes request $i \in P$. Let y_r be a binary decision variable equal to 1 if and only if route $r \in \Omega$ is used in the solution. The PDPTWH can be modeled as

$$\text{minimize} \quad \sum_{r \in \Omega} c_r y_r \tag{5}$$

$$\text{subject to} \quad \sum_{r \in \Omega} a_{ir} y_r = 1, \quad \forall\; i \in P, \tag{6}$$

$$y_r \in \{0, 1\}, \quad \forall\; r \in \Omega. \tag{7}$$

The objective function (5) minimizes the total cost and the set partitioning constraints (6) ensure that each request is completed exactly once.

3. Branch-price-and-cut algorithms

Because model (5)–(7) usually contains a large number of variables we use a branch-price-and-cut algorithm to solve it. A branch-price-and-cut algorithm is a branch-and-cut algorithm that uses column generation to solve the linear relaxations. For the PDPTWH, column generation is used to solve the linear relaxation of model (5)–(7), which is called the master problem. A restricted master problem (RMP), containing only a subset of variables (columns) is solved, yielding a primal and a dual solution. The pricing problem is then solved to identify negative reduced-cost columns with respect to the dual solution. For the PDPTWH, the pricing problem corresponds to an elementary shortest path problem with pickups and deliveries, time windows, one capacity constraint, and rehandling operations. Whenever columns with negative reduced costs are found, they are added to the RMP and a new iteration begins. Otherwise, the process stops with an optimal solution to the master problem. If needed, cutting planes are added to strengthen the linear relaxations and branching is performed to derive integer solutions.

We develop branch-price-and-cut algorithms for the PDPTWH-1 and the PDPTWH-2, where the pricing problems are solved by means of labeling algorithms. We present valid inequalities and the branching strategy for both the PDPTWH-1 and the PDPTWH-2.

3.1. Labeling algorithm for the PDPTWH-1

We propose a labeling algorithm to solve the elementary shortest path problem with pickups and deliveries, a capacity constraint, and time windows under rehandling policy 1. A label E is a resource vector representing a partial path starting at the origin depot and ending at a given node $\eta(E)$. Starting from an initial label E_0 associated with the origin depot and representing an empty path, labels are extended forwardly throughout graph G using resource extension functions to create new labels (paths). The number of generated labels is reduced by eliminating dominated ones with a valid dominance criterion.

A given label E contains the following components:

- $\eta(E)$, the node of the label;
- $t(E)$, the start of the service time and rehandling operations at node $\eta(E)$;
- $l(E)$, the load of the vehicle after visiting node $\eta(E)$;
- $c(E)$, the cumulated reduced cost up to node $\eta(E)$;
- $\mathcal{S}_{ij}(E), \quad \forall\, i, j \in P$, a binary matrix indicating the relative positions between any pair of items i and j in the vehicle before visiting node $\eta(E)$;
- $U(E)$, the set of unreachable requests after visiting node $\eta(E)$.

As defined in Section 1, items are at the same level if the most recent handling operation for both items occurred at the same location. Then, $\mathcal{S}_{ij}(E)$ is given by:

$$\mathcal{S}_{ij}(E) = \begin{cases} 1 & \text{if items } i \text{ and } j \text{ are at the same level,} \\ 1 & \text{if item } i \text{ is on top of item } j, \\ 1 & \text{if } i = j \text{ and item } i \text{ is in the vehicle,} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

For a given label E such that $\eta(E) \in D$ the set of items on top of its corresponding delivery item $\eta(E) - n$ is defined as

$$O_{\eta(E)}(E) = \{ i \in P | \mathcal{S}_{i,\eta(E)-n}(E) = 1 \quad \text{and} \quad \mathcal{S}_{\eta(E)-n,i}(E) = 0 \}. \tag{9}$$

Otherwise, if $\eta(E) \notin D$, $O_{\eta(E)}(E)$ is not defined.

For a given label E, let $R(E)$ be its corresponding partial path, i.e., $R(E) = \{ i_0 = 0, i_1, \ldots, i_{\rho-1}, i_\rho = \eta(E) \}$. Then, the unreachable requests $U(E)$ are defined as the requests for which either the pickup nodes have already been visited on the path, or the extension to the pickup node would violate the time windows, that is

$$U(E) = \begin{cases} \{ i \in P | i \in R(E) \} \cup \{ i \in P | t(E) + s_{\eta(E)} + t_{\eta(E),i} > \overline{w}_i \} & \text{if } \eta(E) \in P, \\ \{ i \in P | i \in R(E) \} \cup \{ i \in P | t(E) + s_{\eta(E)} + \delta \sum_{k \in O_{\eta(E)}(E)} q_k + t_{\eta(E),i} > \overline{w}_i \} & \text{if } \eta(E) \in D. \end{cases}$$

The extension along arc $(\eta(E), j) \in A$ for a given label E is allowed if one of the following four conditions is respected:

$$j \in P \quad \text{and} \quad j \notin U(E), \tag{10}$$

$$j \in D \quad \text{and} \quad \mathcal{S}_{j-n,j-n}(E) = 1, \tag{11}$$

$$j \in D \quad \text{and} \quad \eta(E) = j - n, \tag{12}$$

$$j = 2n + 1, \eta(E) \in D, \quad \text{and} \quad \mathcal{S}_{ii}(E) = 0, \quad \forall\, i \in P \setminus \{ \eta(E) - n \}. \tag{13}$$

Condition (10) ensures that if j is a pickup node, it must be reachable. Conditions (11) and (12) state that if node j is a delivery node, then its corresponding item is onboard. Condition (13) ensures that, if node j is the destination depot, the vehicle is empty.

The reduced cost for arc $(i, j) \in A$ is as follows:

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \alpha_i, & \forall\, i \in P, \\ c_{ij}, & \forall\, i \in N \setminus P, \end{cases} \tag{14}$$

where α_i, $i \in P$, are the dual variables corresponding to constraints (6).

The extension of label E along arc $(\eta(E), j)$ will create a new label E'. The components of this new label are updated as follows:

$$\eta(E') = j, \tag{15}$$

$$t(E') = \begin{cases} \max\{ \underline{w}_j, t(E) + s_{\eta(E)} + t_{\eta(E),j} \} & \text{if } \eta(E) \in P, \\ \max\left\{ \underline{w}_j, t(E) + s_{\eta(E)} + \delta \sum\limits_{i \in O_{\eta(E)}(E)} q_i + t_{\eta(E),j} \right\} & \text{if } \eta(E) \in D, \end{cases} \tag{16}$$

$$l(E') = l(E) + q_j, \tag{17}$$

$$c(E') = c(E) + \bar{c}_{\eta(E),j}, \tag{18}$$

$$U(E') = \begin{cases} U(E) \cup \{j\} \cup \{ i \in P | t(E') + s_j + t_{ji} > \overline{w}_i \} & \text{if } j \in P, \\ U(E) \cup \left\{ i \in P | t(E') + s_j + \delta \sum\limits_{k \in O_j(E')} q_k + t_{ji} > \overline{w}_i \right\} & \text{if } j \in D, \end{cases} \tag{19}$$

$$\mathcal{S}_{ki}(E') = \begin{cases} 1 & \text{if } \eta(E) \in P,\ k = \eta(E), \text{ and } \mathcal{S}_{ii}(E) = 1, \\ 1 & \text{if } \eta(E) \in P \text{ and } k = i = \eta(E), \\ 1 & \text{if } \eta(E) \in D \text{ and } k, i \in O_{\eta(E)}(E), \\ 0 & \text{if } \eta(E) \in D \text{ and } i = \eta(E) - n, \\ 0 & \text{if } \eta(E) \in D \text{ and } k = \eta(E) - n, \\ \mathcal{S}_{ki}(E) & \text{otherwise.} \end{cases} \quad \forall\, k \in P, \forall\, i \in P \tag{20}$$

Eqs. (20) update the relative positions of the items upon leaving node $\eta(E)$. If $\eta(E)$ is a pickup node, item $\eta(E)$ is onboard and positioned on top of all other onboard items. If $\eta(E)$ is a delivery node, the set of items that are on top of item $\eta(E) - n$ are rehandled, implying that all those items are at the same level, and item $\eta(E) - n$ is unloaded from the vehicle. All other positions remain unchanged.

Label E' is kept if the time windows and the capacity constraint are respected, that is if

$$t(E') \le \overline{w}_j, \tag{21}$$

$$l(E') \le Q. \tag{22}$$

A label E_1 dominates label E_2 if

$$\eta(E_1) = \eta(E_2), \tag{23}$$

$$t(E_1) \le t(E_2), \tag{24}$$

$$l(E_1) \le l(E_2), \tag{25}$$

$$c(E_1) \le c(E_2), \tag{26}$$

$$U(E_1) \subseteq U(E_2), \tag{27}$$

$$\mathcal{S}_{ij}(E_1) = \mathcal{S}_{ij}(E_2), \quad \forall\, i, j \in P. \tag{28}$$

Note that conditions (28) imply that the same items must be onboard the vehicle in the paths associated with labels E_1 and E_2.

Proposition 1. *Conditions (23)–(28) constitute a valid dominance criterion for the elementary shortest path problem with pickups and deliveries, a capacity constraint, and time windows under rehandling policy 1.*

Proof. We show that every feasible completion of E_2 is also feasible for E_1 and yields a reduced cost that is no worse than when it completes the path associated with E_2. Let $R(E)$ as previously defined denote the partial path corresponding to label E. Let r be a path extending $R(E_2)$ to node $2n+1$ such that $(R(E_2), r)$ is feasible with respect to the elementarity constraints, pickup and delivery constraints, capacity constraint, and time windows under rehandling policy 1. If no such path exists, label E_2 can be removed. Because $(R(E_2), r)$ is feasible with respect to the elementary constraints, pickup and delivery constraints, and capacity constraint, then so is $(R(E_1), r)$. The time windows are not violated under rehandling policy 1, because the set of onboard items for label E_2 is equivalent to the set of onboard items for label E_1 and their relative positions are the same, that is $S_{ij}(E_1) = S_{ij}(E_2),\ \forall\, i, j \in P$. Because $(R(E_2), r)$ is feasible with respect to the time windows under rehandling policy 1, then so is $(R(E_1), r)$. Because $c(E_1) \le c(E_2)$, the cost of $(R(E_1), r)$ is at most equal to that of $(R(E_2), r)$. Hence, label E_1 dominates label E_2. □

We cannot easily relax condition (28). In Figs. 4 and 5, we give two examples that illustrate the underlying difficulties. We show that comparing items that are not at the same level might lead to wrongly dominated labels. Both examples compare two labels E_1 and E_2, where a feasible extension of label E_2 may not be feasible for label E_1. The same items are onboard the vehicle corresponding to labels E_1 and E_2, but two items that are at the same level in the vehicle of one label are at different levels in the vehicle of the other label. In the first example, two items, i.e., items 1 and 2, are at the same level in the vehicle of label E_2, but on different levels in the vehicle of label E_1. For label E_1, item 2 is rehandled upon delivering item 1, whereas for label E_2, item 2 can be delivered without rehandling. Therefore, the time window at node 2^- could be respected for label E_2 but not for label E_1. In the second example, two items, i.e., items 1 and 2, are at the same level in the vehicle of label E_1, but on different levels in the vehicle of label E_2. For label E_1, item 3 is rehandled upon delivering item 1, whereas item 2 cannot be rehandled since policy 1 only allows for compulsory rehandling. Therefore, item 3 is rehandled again upon delivering item 2. For label E_2, items 2 and 3 are rehandled upon delivering item 1, and there is no rehandling upon delivering item 2. Therefore, the time window at node 3^- could be respected for label E_2 but not for label E_1.

3.2. Labeling algorithm for the PDPTWH-2

We propose a labeling algorithm for the elementary shortest path problem with pickups and deliveries, a capacity constraint, and time windows under rehandling policy 2. In a given label E, the components $\eta(E)$, $t(E)$, $l(E)$, $c(E)$, $S_{ij}(E)$, and $U(E)$ defined as in Section 3.1 are stored. The extension along arc $(\eta(E), j) \in A$ for a given label E is allowed if one of the conditions (10)–(13) is respected.

For label E, for each onboard item $i \in P$ at node $\eta(E)$, we define $\mathcal{B}_i(E) = \{j \in P | S_{ij}(E) = 1 \text{ and } S_{ji}(E) = 0\}$ as the set of items below i, $\mathcal{L}_i(E) = \{j \in P \setminus \{i\} | S_{ij}(E) = S_{ji}(E) = 1\}$ as the set of items at the same level as i, and $\mathcal{T}_i(E) = \{j \in P | S_{ij}(E) = 0 \text{ and} S_{ji}(E) = 1\}$ as the set of items on top of i. Thus, under policy 2, for a given label E such that $\eta(E) \in D$, all items that need compulsory rehandling are in $\mathcal{T}_{\eta(E)-n}(E)$. Preventive rehandling is allowed for the items at the same level, i.e., in $\mathcal{L}_{\eta(E)-n}(E)$, and the items below $\eta(E)-n$, i.e., in $\mathcal{B}_{\eta(E)-n}(E)$, as long as all items that are on top are handled. Let $\mathcal{H}(E)$ represent the set of all feasible combinations of rehandled items. For $\eta(E) \in P$, we have $\mathcal{H}(E) = \varnothing$. For $\eta(E) \in D$, $\mathcal{H}(E)$ is the collection of all feasible combinations of rehandled items, where a feasible combination of rehandled items is a set $S \subseteq \mathcal{B}_{\eta(E)-n}(E) \cup \mathcal{L}_{\eta(E)-n}(E) \cup \mathcal{T}_{\eta(E)-n}(E)$ such that, if $i \in S, j \in \mathcal{T}_i(E)$, and $j \neq \eta(E)-n$, then $j \in S$, and if $k \in \mathcal{T}_{\eta(E)-n}(E)$, then $k \in S$, i.e., if an item is in a feasible set of rehandled items, then all items on top (except for the delivered item) are also in this set, and so are all items on top of the delivered item.

Fig. 6 illustrates a vehicle configuration for a given label E corresponding to the partial path $R(E) = \{0, \ldots, k, 3^-\}$. In this example, we have $\mathcal{T}_{\eta(E)-n}(E) = \{1\}$, $\mathcal{L}_{\eta(E)-n}(E) = \{2, 4\}$, $\mathcal{B}_{\eta(E)-n}(E) = \{5, 6, 7\}$. The set of all feasible combinations of rehandled items is given by $\mathcal{H}(E) = \{\{1\}, \{1, 2\}, \{1, 4\}, \{1, 2, 4\}, \{1, 2, 4, 5\}, \{1, 2, 4, 6\}, \{1, 2, 4, 5, 6\}, \{1, 2, 4, 5, 6, 7\}\}$.

The extension of label E along arc $(\eta(E), j)$ creates multiple labels E^h, $\forall\, h \in \{1, \ldots, |\mathcal{H}(E)|\}$, that is one for each feasible combination of rehandled items. Let $\mathcal{H}^h(E)$ be the set of rehandled items corresponding to label extension $h \in \{1, \ldots, |\mathcal{H}(E)|\}$. The components of label E^h are set as follows:

$$\eta(E^h) = j, \tag{29}$$

$$t(E^h) = \begin{cases} \max\{\underline{w}_j, t(E) + s_{\eta(E)} + t_{\eta(E),j}\} & \text{if } \eta(E) \in P, \\ \max\left\{\underline{w}_j, t(E) + s_{\eta(E)} + \delta \sum_{i \in \mathcal{H}^h(E)} q_i + t_{\eta(E),j}\right\} & \text{if } \eta(E) \in D, \end{cases} \tag{30}$$

$$l(E^h) = l(E) + q_j, \tag{31}$$

$$c(E^h) = c(E) + \bar{c}_{\eta(E),j}, \tag{32}$$

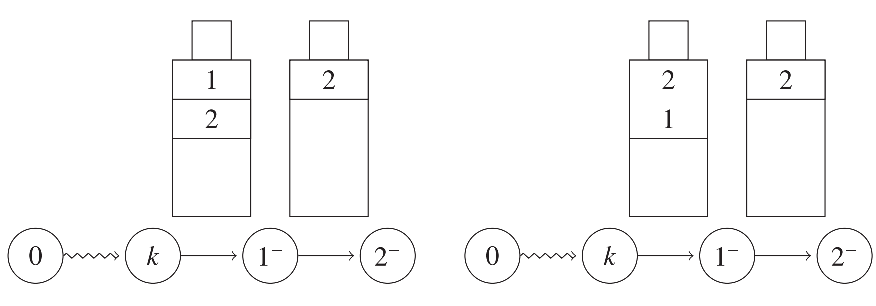

Fig. 4. Possible extension of labels $E_1$and E_2, where items 1 and 2 are at the same level for label E_2, but at different levels for label E_1. The extension might be feasible for label E_2 but not for label E_1. (a) Partial route and vehicle configurations corresponding to label E_1, where $\eta(E_1) = 1^-$, $S_{11}(E_1) = S_{22}(E_1) = S_{21}(E_1) = 1$, and all other $S_{ij}(E_1) = 0$. (b)Partial route and vehicle configurations corresponding to label E_2, where $\eta(E_2) = 1^-$, $S_{11}(E_2) = S_{22}(E_2) = S_{12}(E_2) = S_{21}(E_2) = 1$, and all other $S_{ij}(E_2) = 0$.

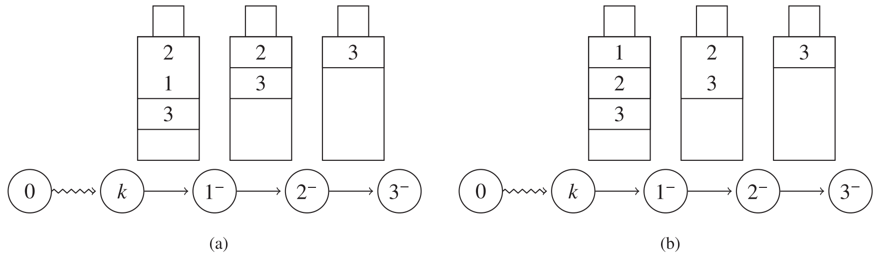

Fig. 5. Possible extension of labels E_1 and E_2, where items 1 and 2 are at the same level for label E_1, but at different levels for label E_2. The extension might be feasible for label E_2 but not for label E_1. (a) Partial route and vehicle configurations corresponding to label E_1, where $\eta(E_1) = 1^-$, $\mathcal{S}_{11}(E_1) = \mathcal{S}_{22}(E_1) = \mathcal{S}_{33}(E_1) = \mathcal{S}_{12}(E_1) = \mathcal{S}_{21}(E_1) = \mathcal{S}_{32}(E_1) = \mathcal{S}_{31}(E_1) = 1$, and all other $\mathcal{S}_{ij}(E_1) = 0$. (b) Partial route and vehicle configurations corresponding to label E_2, where $\eta(E_2) = 1^-$, $\mathcal{S}_{11}(E_2) = \mathcal{S}_{22}(E_2) = \mathcal{S}_{33}(E_2) = \mathcal{S}_{21}(E_2) = \mathcal{S}_{31}(E_2) = \mathcal{S}_{32}(E_2) = 1$, and all other $\mathcal{S}_{ij}(E_2) = 0$.

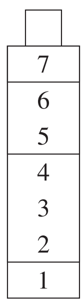

Fig. 6. A vehicle configuration corresponding to label E associated with node $\eta(E) = 3^-$.

$$U(E^h) = \begin{cases} U(E) \cup \{j\} \cup \{i \in P | t(E^h) + t_{\eta(E^h),i} + s_{\eta(E^h)} > \overline{w}_i\} & \text{if } j \in P, \\ U(E) \cup \left\{ i \in P | t(E^h) + t_{\eta(E^h),i} + s_{\eta(E^h)} + \delta \sum_{k \in \mathcal{H}^h(E^h)} q_k > \overline{w}_i \right\} & \text{if } j \in D, \end{cases} \tag{33}$$

$$\mathcal{S}_{ki}(E^h) = \begin{cases} 1 & \text{if } \eta(E) \in P,\ k = \eta(E), \quad \text{and} \quad \mathcal{S}_{ii}(E) = 1, \\ 1 & \text{if } \eta(E) \in P \quad \text{and} \quad k = i = \eta(E), \\ 1 & \text{if } \eta(E) \in D \quad \text{and} \quad k, i \in \mathcal{H}^h(E), \\ 0 & \text{if } \eta(E) \in D,\ \mathcal{S}_{kk}(E) = 1,\ k \neq \eta(E) - n,\ k \notin \mathcal{H}^h(E), \quad \text{and} \quad i \in \mathcal{H}^h(E),\ \forall\, k, i \in P. \\ 0 & \text{if } \eta(E) \in D \quad \text{and} \quad i = \eta(E) - n, \\ 0 & \text{if } \eta(E) \in D \quad \text{and } k = \eta(E) - n, \\ \mathcal{S}_{ki}(E) & \text{otherwise,} \end{cases} \tag{34}$$

that all items that are on top of each other in the vehicle corresponding to label E_1 must also be on top of each other in the vehicle corresponding to label E_2, and all items that are at the same level in E_1 can be or not at the same level in the vehicle corresponding to label E_2.

Proposition 2. *Conditions (23)–(27), (35) and (36) constitute a valid dominance criterion for the elementary shortest path problem with pickups and deliveries, a capacity constraint, and time windows under rehandling policy 2.*

Proof. We show that every feasible completion of E_2 is also feasible for E_1 and yields a reduced cost that is no worse than when it completes the path associated with E_2. Let $R(E)$ as previously defined denote the partial path corresponding to label E. Let r be a path extending $R(E_2)$ to node $2n + 1$ such that $(R(E_2), r)$ is feasible with respect to the elementarity constraints, pickup and delivery constraints, capacity constraint, and time windows under rehandling policy 2. If no such path exists, label E_2 can be removed. Because $(R(E_2), r)$ is feasible with respect to the elementarity constraints, pickup and delivery constraints, and capacity constraint, then so is $(R(E_1), r)$. Conditions (35) and (36) imply that labels E_1 and E_2 have the same set of onboard items. In addition, condition (36) ensures that all items that are at the same level for label E_2 are also at the same level for label E_1. Items that are not at the same level for E_2 either have the same relative positions for label E_1 or are at the same level for label E_1. In the latter case, the items can be ordered similarly to the items in label E_2, since all items can be rehandled at the delivery nodes. Hence, at each delivery node, the number of rehandling operations for label E_1 cannot be more than for label E_2, thus $(R(E_1), r)$ is feasible with respect to the time windows under policy 2. Because $(R(E_2), r)$ is

Eq. (34) updates the relative positions of the items upon leaving $\eta(E)$. If $\eta(E)$ is a pickup node, item $\eta(E)$ is positioned on top of all other onboard items. If $\eta(E)$ is a delivery node, all items in the set $\mathcal{H}^h(E)$ are rehandled, implying that all those items are at the same level and on top of the other onboard items. Moreover, item $\eta(E) - n$ is unloaded from the vehicle. All other positions remain unchanged. Label E^h is kept if the time windows and the capacity constraint are respected, that is if inequalities (21) and (22) are respected. A label E_1 dominates a label E_2 if it respects (23)–(27) and

$$\mathcal{S}_{ij}(E_1) \leq \mathcal{S}_{ij}(E_2)\ \forall\, i, j \in P \text{ such that } \mathcal{S}_{ij}(E_1) + \mathcal{S}_{ji}(E_1) \leq 1 \quad \text{or } i = j, \tag{35}$$

$$\mathcal{S}_{ij}(E_1) + \mathcal{S}_{ji}(E_1) = 2 \quad \forall\, i, j \in P \text{ such that } \mathcal{S}_{ij}(E_2) + \mathcal{S}_{ji}(E_2) = 2. \tag{36}$$

Conditions (35) and (36) imply that E_1 and E_2 have the same set of onboard items, i.e., $\mathcal{S}_{ii}(E_1) = \mathcal{S}_{ii}(E_2), \quad \forall\, i \in P$. Moreover, they imply feasible, then so is $(R(E_1), r)$. Because $c(E_1) \leq c(E_2)$, the cost of $(R(E_1), r)$ is at most equal to that of $(R(E_2), r)$. Hence, label E_1 dominates label E_2. □

3.3. Valid inequalities

We now present the valid inequalities used in the branch-price-and-cut algorithm for both the PDPTWH-1 and the PDPTWH-2. These are valid inequalities based on the number of vehicles, the subset-row inequalities, and the two-path cut inequalities.

As in Cherkesly et al. [5], we propose valid inequalities based on the number of vehicles to generate a solution with an integer number of vehicles. If the linear relaxation solution $(\tilde{y}_1, \ldots, \tilde{y}_{|\Omega|})$ at a node of the search tree involves a fractional number of vehicles, we round up this number by adding the inequality

$$\sum_{r\in\Omega} y_r \geq \left\lceil \sum_{r\in\Omega} \tilde{y}_r \right\rceil. \tag{37}$$

This inequality is valid assuming that the vehicle fixed cost is sufficiently large to minimize first the number of vehicles used.

Jepsen et al. [11] introduced the subset-row inequalities for the vehicle routing problem with time windows (VRPTW). For the PDPTWH, these inequalities are given by

$$\sum_{r\in\Omega} \left\lfloor \frac{1}{\chi} \sum_{i\in S} a_{ir} \right\rfloor y_r = \left\lfloor \frac{|S|}{\chi} \right\rfloor, \quad \forall\ S \subseteq P,\ 2 \leq \chi \leq |S|, \tag{38}$$

where $S \subseteq P$ is a subset of pickup nodes. As in Cherkesly et al. [5], Desaulniers et al. [9], and Jepsen et al. [11], we consider the inequalities imposed by subsets of three customers. These inequalities can be rewritten as

$$\sum_{r\in\Omega_S} y_r \leq 1, \quad \forall\ S \subseteq P \text{ such that } |S| = 3, \tag{39}$$

where $\Omega_S \subseteq \Omega$ is the subset of paths that complete at least two requests in S. These inequalities are separated by means of an enumerative procedure.

The two-path inequalities were introduced by Kohl et al. [12] for the VRPTW and adapted for several other vehicle routing problems including the PDPTW and the PDPTWL (Ropke and Cordeau [16]; Cherkesly et al. [5]). Let $S \subseteq P \cup D$ be a subset of nodes that needs to be served by more than one vehicle because of the time windows. Then, the inequality

$$\sum_{r\in\Omega} \sum_{(i,j)\in A|i\in S, j\notin S} b_{ij}^r y_r \geq 2, \tag{40}$$

is valid, where b_{ij}^r is the number of times arc $(i, j) \in A$ is used in path $r \in \Omega$. A set $S \subseteq P \cup D$ needs to be served by more than one vehicle if there is no feasible path visiting all nodes in S. Since identifying such a set S is NP-complete, a greedy heuristic is used to separate these valid inequalities (Ropke and Cordeau [16]).

The order in which the cuts are generated is as follows: (1) the valid inequalities based on the number of vehicles, i.e., inequalities (37), (2) the two-path cut inequalities, i.e., inequalities (40), and (3) the subset-row inequalities, i.e., inequalities (39). Those inequalities are added to the master problem and the associated duals modify the arc reduced costs. In order to account for the duals of inequalities (38), the labeling algorithm must be modified (see Desaulniers et al. [8]).

3.3.1. Acceleration strategies for the labeling algorithms

In order to speed up the labeling algorithms, a heuristic is used for the dominance under policies 1 and 2: we start with a reduced network and slowly increase the size of the network to its full size, thus ensuring that the solution is optimal. Moreover, for policy 1 we use a heuristic dominance criterion which corresponds to Eqs. (23)–(27), (35) and (36). When no more columns of negative reduced costs are found, the exact dominance criterion which corresponds to Eqs. (23)–(28) is applied to ensure that the algorithm ends up with an optimal solution.

3.4. Branching

Branching is imposed to obtain an integer feasible solution. In branch-and-price, the branching strategies must be consistent with the column generation approach, especially with the labeling algorithm. For the PDPTWH-1 and the PDPTWH-2, two different branching strategies can be applied; branching on arcs and branching on the outflow of a subset of nodes. The branch-and-bound search tree is explored through a best-first strategy.

Branching on arcs was proposed by Christofides et al. [7] for the CVRP. The arc $(i, j) \in A$, $i \neq 0$, $j \neq 2n + 1$, with the largest difference of $\sum_{r\in\Omega} b_{ij}^r \tilde{y}_r$ to the nearest integer is selected. Two branches are then created:

$$\sum_{r\in\Omega} b_{ij}^r y_r = 0, \tag{41}$$

$$\sum_{r\in\Omega} b_{ij}^r y_r = 1. \tag{42}$$

For both branches, the underlying network is adapted, and therefore no complexity is added. For the first branch, the selected arc is removed from the network, and for the second branch, all arcs $(i, k) \in A$ such that $k \neq j$ and all arcs $(k, j) \in A$ such that $k \neq i$ are removed.

Naddef and Rinaldi [13] proposed branching on the outflow of a subset of nodes for the CVRP. This has later been adapted by Ropke and Cordeau [16] for the PDPTW, and used by Cherkesly et al. [5] for the PDPTWL. The subset of nodes S with the largest difference of $\sum_{r\in\Omega} \sum_{(i,j)\in A|i\in S, j\notin S} b_{ij}^r \tilde{y}_r$ to the nearest integer is selected. For this set of nodes, two branches are created:

$$\sum_{r\in\Omega} \sum_{(i,j)\in A|i\in S, j\notin S} b_{ij}^r y_r \leq \left\lfloor \sum_{r\in\Omega} \sum_{(i,j)\in A|i\in S, j\notin S} b_{ij}^r \tilde{y}_r \right\rfloor, \tag{43}$$

$$\sum_{r\in\Omega} \sum_{(i,j)\in A|i\in S, j\notin S} b_{ij}^r y_r \geq \left\lceil \sum_{r\in\Omega} \sum_{(i,j)\in A|i\in S, j\notin S} b_{ij}^r \tilde{y}_r \right\rceil. \tag{44}$$

Inequalities (43) and (44) are added to the master problem and their dual variables influence the arc reduced costs.

Preliminary experiments showed that branching on the outflow of a subset of nodes outperformed branching on arcs or a combination of both strategies. Therefore, we only used branching on the outflow of a subset of nodes in the algorithms for the PDPTWH-1 and the PDPTWH-2. In the branch-and-cut loop, priority is given to the cuts.

4. Computational results

The branch-price-and-cut algorithms were implemented using the GENCOL library and CPLEX 12.4.0.0 to solve the restricted master problems. All experiments were performed on a Linux computer equipped with an Intel(R) Core(TM) i7-3770 processor (3.4 GHz). We have tested our algorithms on a modified version of the instances of Ropke and Cordeau [16]. We will discuss the impact of the rehandling policies and the effect of the parameter value for the rehandling time on the results. Moreover, we compare our results with those obtained by Cherkesly et al. [5] for the PDPTWL.

4.1. Instances

To test the algorithms, we used the modified instances of Ropke and Cordeau [16] as described by Cherkesly et al. [5]. The instance set is composed of four groups AA, BB, CC, and DD, each containing 10 instances, where the number of requests ranges from 30 to 75. Moreover, we created the two additional instance groups AA′ and BB′, by increasing the vehicle capacity in the AA and BB instances. The characteristics of the instances are summarized in Table 1. This table reports for each group of instances the vehicle capacity Q, the width of the time windows W, the delay of the delivery nodes Δ, and the average demand per customer d. The parameter Δ is used to adjust the time windows of the delivery nodes with respect to the original time windows of Ropke and Cordeau [16] as follows $\underline{w}_i = \underline{w}_i + \Delta$ and $\overline{w}_i = \overline{w}_i + \Delta, \ \forall \ i \in D$. This adjustment is performed to increase the number of requests that can be onboard the same vehicle simultaneously. A fixed vehicle cost equal to 10,000 is added to each arc $(0, i)$, $i \in P$, to ensure the minimization of the vehicles as a primary objective. We consider several values for the rehandling parameter δ, namely $\delta \in \{0, 0.5, 1, 5, 10, 50\}$.

Table 1
Instance characteristics.

Group	Q	W	Δ	d
AA	22	60	45	9.9
BB	30	60	45	13.2
CC	18	120	15	10.0
DD	25	120	15	11.8
AA′	26	60	45	9.9
BB′	35	60	45	13.2

4.2. Summary of the results

The branch-price-and-cut algorithms for the PDPTWH-1 and the PDPTWH-2 were run with a time limit of three hours for each instance. Tables 2 and 3 provide a summary of the results. For each group of instances and each value of δ, these tables report #, the number of instances solved to optimality within the specified time limit, *Sec.*, the average computation time in seconds, and *Gap*, the average optimality gap in percentage. The optimality gap is computed as $(z^* - \underline{z})/\underline{z}$, where z^* is the optimal value and $\underline{z}$ is the lower bound obtained at the root node before adding cuts. The average computation time and the average optimality gap are computed by only including the instances that could be solved to optimality within the time limit. Detailed computational results are reported in the Appendix.

Comparing the results for the PDPTWH-1 with those for the PDPTWH-2, we observe that some instances were solved to optimality within the prescribed time limit by the algorithm for the PDPTWH-1, but not for the PDPTWH-2, i.e., instances AA70 with $\delta = 1$, CC55 with $\delta = 5$, DD30 with $\delta = 0$, BB′55 with $\delta = 1$, and BB′ 60 with $\delta = 1$. On the other hand, the three instances CC55 with $\delta = 1$, AA′60 with $\delta = 0.5$, and BB′70 with $\delta = 5$ were solved to optimality by the algorithm for the PDPTWH-2, but not for the PDPTWH-1. Comparing the results for the different instance groups, we see that increasing the width of the time windows makes the problem harder to solve. For the AA and BB groups, the total number of instances for all rehandling time δ solved within the time limit is 56 and 52 for the PDPTWH-1, respectively, and 55 and 52 for the PDPTWH-2, respectively. For the CC and DD groups fewer instances are solved, i.e., 20 and 4 for the PDPTWH-1, respectively, and 20 and 3 for the PDPTWH-2, respectively. Increasing the width of the time windows results in harder to solve elementary shortest path problems, which probably explains the large amount of time needed for the instances in the CC and DD groups. Having a closer look at the effect of the capacity increase, we observe that fewer instances were solved to optimality within the prescribed time limit for the AA’ group compared with the AA group, namely 41 compared to 56 for the PDPTWH-1, and 42 compared to 55 for the PDPTWH-2. A similar result can be found when comparing the BB’ group with the BB group, namely 42 compared to 52 instances that were solved to optimality with the prescribed time limit for the PDPTWH-1, and 41 compared to 52 for the PDPTWH-2. Again, by increasing the vehicle capacity, the elementary shortest path problems are harder to solve.

By increasing the value of δ, we see that the total number of instances solved to optimality within the time limit first decreases and then increases. With a value of $\delta = 50$ most instances were solved, namely 66.7% of all instances. This pattern holds for both the PDPTWH-1 and the PDPTWH-2. There are four exceptions: instance groups CC and BB’ for the PDPTWH-1 and instance groups CC and DD for the PDPTWH-2. For the PDPTWH-1 and $\delta \in \{0, 5\}$, and for the PDPTWH-2 and $\delta \in \{0, 1\}$ four instances were solved to optimality in the CC group and three instances were solved to optimality for the other values of δ. For the PDPTWH-1 and group BB′, and for the PDPTWH-2 and group DD we only observe an increase and no decrease in the number of instances solved to optimality within the time limit.

4.3. Impact of the rehandling policy

Table 4 provides a comparison between the travel costs for the optimal solutions under policies 1 and 2. We only report the instances for which the value of the optimal solution differs between the two rehandling policies, namely BB35, BB45, BB50, BB55, AA′40, AA′45, AA′50, AA′55, BB′55, and BB′60. For all these instances, the algorithms for the PDPTWH-1 and the PDPTWH-2 provide optimal solutions with the same costs for $\delta \in \{0, 10, 50\}$. Therefore, we only report the results for $\delta \in \{0.5, 1, 5\}$. For each of these values of δ the table reports *TC1* and *TC2*, which are the travel costs corresponding to the optimal solutions obtained for the PDPTWH-1 and PDPTWH-2, respectively, and *Diff* (%), the difference in percentage between the travel costs corresponding to the optimal solutions of policies 1 and 2 computed as $(TC1 - TC2)/TC2$.

For the 10 instances in Table 4 and for various values of δ, the value of the optimal solution differs under the two policies. For these instances, the number of vehicles used in the optimal solution is the same under both rehandling policies and the difference in costs is due to the difference in travel costs. These instances are all contained in the BB, AA′, or BB′ group, which are the groups with the largest vehicle capacity. This suggests that the extra flexibility in rehandling operations can be more effectively used when the vehicle capacity is larger. Because more items can be onboard a vehicle if its capacity increases, this results in more options for preventive rehandling operations. The decrease in travel costs gained by allowing more flexibility in the rehandling operations is up to 3.2% for the instances under study.

4.4. Impact of the rehandling time

We now analyze the impact of the rehandling time by comparing the values of the optimal solutions for the PDPTWH for different values of δ. Since the values of the optimal solutions for the instances under study are quite comparable for rehandling policies 1 and 2, we compare the results under the least restrictive policy, which is rehandling policy 2. Similar results can be obtained for rehandling policy 1. Note that, for the instances under study, the optimal solutions for the PDPTWH-2 with $\delta = 50$ correspond to the optimal solutions for the PDPTWL.

Table 5 presents the increase in travel costs and number of

Table 2
Summary results for the PDPTWH-1.

Group	$\delta = 0$			$\delta = 0.5$			$\delta = 1$			$\delta = 5$			$\delta = 10$			$\delta = 50$		
	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap
AA	10	1292.9	3.6	8	168.4	2.6	9	1346.0	4.0	9	480.5	3.7	10	309.8	4.2	10	239.5	4.2
BB	8	380.1	3.1	8	1486.8	7.7	6	464.0	6.1	10	755.8	5.5	10	225.2	5.5	10	357.5	5.5
CC	4	2700.2	26.4	3	78.6	27.0	3	106.1	26.8	4	2173.3	23.2	3	67.9	26.6	3	52.9	26.6
DD	1	3346.3	15.0	0			0			1	6331.4	42.0	1	8699.9	42.1	1	7030.5	42.1
AA′	8	258.0	3.9	6	1003.6	5.2	5	916.8	6.2	7	1617.2	3.9	7	238.9	1.9	8	848.5	1.7
BB′	6	505.7	5.1	6	846.1	11.4	6	2921.2	14.1	7	1043.3	9.2	8	1344.6	8.9	9	866.1	8.3

vehicles used when increasing the rehandling time δ. The first column reports the name of the instances defined as its group and the number of requests. For $\delta = 0$, we report *Veh.*, the number of vehicles used in the optimal solution and *TC*, the total travel costs in this solution. For all other values of δ, we report $\Delta Veh.$, the increase in the number of vehicles computed as $\text{Veh}_\delta - \text{Veh}_0$, where Veh_δ and Veh_0 are the number of vehicles in the optimal solution with $\delta \in \{0.5, 1, 5, 10, 50\}$ and $\delta = 0$, respectively, and $\Delta TC\ (\%)$, the impact in percentage on the total travel costs computed as $(\text{TC}_\delta - \text{TC}_0)/\text{TC}_0$, where TC_δ and TC_0 are the travel costs in the optimal solution with $\delta \in \{0.5, 1, 5, 10, 50\}$ and $\delta = 0$, respectively. We only report instances for which the PDPTWH-2 with $\delta = 0$ has been solved within the prescribed time limit.

The primary objective is to minimize the number of vehicles used in the solution. By increasing the value of δ, the optimal solution for the problem with $\delta = 0$ may not be feasible anymore with respect to the time windows. Therefore, it occurs for some instances that for a value of $\delta > 0$, all feasible solutions use more vehicles than the optimal solution with $\delta = 0$. If so, the number of vehicles could increase, but the travel costs could decrease. This is actually the case for the following instances: AA65, BB40, BB55, BB′40, and BB′45. For nine instances, i.e., AA55, AA60, AA65, BB40, BB55, AA′50, BB′40, BB′45, and BB′55, the number of vehicles used in the optimal solution increases, compared with the optimal solution with $\delta = 0$. For the five instances AA65, BB40, BB55, BB′40, and BB′45, this increase is already observed for $\delta = 0.5$, for the other four instances the increase in vehicles used in the optimal solution is observed for the first time for $\delta = 5$. For the instances under study, the travel costs can increase up to 24.8%. For all instances in the groups AA′ and BB′, the increase in rehandling time has a large impact on the value of the optimal solutions. Comparing the value of the optimal solutions for $\delta = 0$ and $\delta = 50$, we observe that for each of these instances either the number of vehicles used in the optimal solution increases, or the travel costs increase with at least 10.7%.

4.5. Comparison with the PDPTWL

The optimal solutions for the PDPTWH-1 and the PDPTWH-2 with $\delta = 50$, correspond to the optimal solutions for the PDPTWL. In order to gain some insights into the computational complexity added by allowing rehandling operations, we can compare the computation time needed to solve the instances for the three different problems. For the PDPTWL, we use the results obtained by the best branch-price-and-cut algorithm as reported by Cherkesly [3]. The results in Cherkesly [3] are an improvement of the results in Cherkesly et al.[5], obtained by an improved labeling algorithm. Note that the results in Cherkesly [3] are obtained by using the same computer as the one used for the experiments in this paper.

For each instance group AA, BB, CC, and DD, Table 6 reports for the PDPTWL #, the number of instances solved to optimality within the prescribed time limit, and *Sec.*, the average computation time in seconds, and for the PDPTWH-1 and the PDPTWH-2 it reports #, the number of instances solved to optimality within the prescribed time limit, and $\Delta Sec.(\%)$, the increase in percentage of the average computation time, computed as $(\text{Sec}_\text{i} - \text{Sec}_\text{L})/\text{Sec}_\text{L}$, where Sec_i and Sec_L are the average computation times in seconds for the PDPTWH-*i*, $i \in \{1, 2\}$, and the PDPTWL, respectively. Furthermore, we report the row *Total*, which combines all instances solved for groups AA, BB, CC, and DD, and where the time is an average of all instances solved within the prescribed time limit. Except for instances CC50, CC55, DD30, DD40, DD45, and DD50, which are only solved to optimality within the time limit by the algorithm for the PDPTWL, the same instances in groups AA, BB, CC, and DD, could be solved to optimality by the algorithms for the three different problems. In order to make a fair comparison between the computation times for the different problems, we did not include instances CC50, CC55, DD30, DD40, DD45, and DD50 when calculating the average computation time for the PDPTWL. The AA′ and BB′ groups are excluded from the table, because they were not solved by Cherkesly [3] for the PDPTWL.

All instances in the AA and BB groups were solved to optimality within the time limit by all three algorithms. For each of the algorithms, the average computation time was higher for the BB group compared to the AA group. However, comparing the average computation time for the PDPTWH-1 and the PDPTWH-2 with the PDPTWL, we observe that the difference is higher for the AA instances, i.e., 59.6% and 122.3% for the PDPTWH-1 and the PDPTWH-2, respectively, than for the BB instances, i.e., 24.8% and 57.5% for the PDPTWH-1 and the PDPTWH-2, respectively. The

Table 3
Summary results for the PDPTWH-2.

Group	$\delta = 0$			$\delta = 0.5$			$\delta = 1$			$\delta = 5$			$\delta = 10$			$\delta = 50$		
	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap	#	Sec.	Gap
AA	10	1015.1	3.6	8	298.4	2.6	8	646.3	2.3	9	497.0	3.7	10	341.9	4.2	10	333.7	4.2
BB	8	594.6	3.1	8	1705.8	7.8	6	1117.2	6.1	10	852.9	5.5	10	248.6	5.5	10	451.2	5.5
CC	4	2680.5	26.4	3	290.6	27.0	4	1533.1	23.7	3	125.6	26.6	3	69.4	26.6	3	54.1	26.6
DD	0			0			0			1	7012.6	42.0	1	9384.5	42.1	1	7996.8	42.1
AA′	8	1089.8	3.9	7	657.4	4.4	5	594.3	6.2	7	985.8	3.9	7	268.0	1.9	8	1193.6	1.7
BB′	6	325.0	5.1	6	869.5	11.7	4	235.5	11.6	8	1482.2	8.5	8	1832.8	8.9	9	1147.2	8.3

Table 4
Difference in travel costs for the optimal solutions under policies 1 and 2.

Inst.	$\delta = 0.5$			$\delta = 1$			$\delta = 5$		
	TC1	TC2	Diff. (%)	TC1	TC2	Diff. (%)	TC1	TC2	Diff. (%)
BB35	1258.1	1256.7	0.1	1277.8	1277.8	0.0	1310.8	1310.8	0.0
BB45	1413.2	1407.2	0.4	1441.4	1436.5	0.3	1519.6	1519.6	0.0
BB50	1636.4	1625.6	0.7	1685.9	1671.6	0.9	1777.2	1777.2	0.0
BB55	1692.3	1689.5	0.2				1852.4	1846.6	0.3
AA′40	1217.9	1217.9	0.0	1230.8	1229.6	0.1	1343.5	1343.5	0.0
AA′45	1387.3	1384.6	0.2	1414.8	1413.6	0.1	1587.1	1587.1	0.0
AA′50	1526.8	1524.6	0.1	1647.6	1596.1	3.2	1521.2	1521.2	0.0
AA′55	1518.5	1515.8	0.2				1706.6	1706.6	0.0
BB′55	1661.5	1661.5	0.0	1570.3			1776.4	1771.5	0.3
BB′60				1924.2			2073.8	2068.6	0.3

Table 5
Impact on travel costs and number of vehicles used when increasing the rehandling time δ.

Inst.	δ=0		δ=0.5		δ=1		δ=5		δ=10		δ=50	
	Veh.	TC	ΔVeh.	ΔTC (%)	ΔVeh.	ΔTC (%)	ΔVeh.	ΔTC (%)	ΔVeh.	ΔTC (%)	ΔVeh.	ΔTC (%)
AA30	3	969.4	0	2.0	0	2.2	0	10.1	0	16.5	0	16.5
AA35	3	1089.0	0	0.1	0	2.3	0	13.0	0	18.8	0	18.8
AA40	4	1241.7	0	1.0	0	2.2	0	6.8	0	8.4	0	8.7
AA45	4	1412.2	0	0.7	0	2.2	0	6.6	0	7.5	0	7.7
AA50	4	1531.6	0	0.3	0	1.8	0	6.7	0	7.3	0	7.3
AA55	4	1667.1	0	1.8	0	5.9	1	2.0	1	4.6	1	4.6
AA60	4	1822.7	0	1.4	0	5.4	1	4.0	1	6.9	1	7.0
AA65	4	2011.6	1	−6.2	1	−5.6	1	0.5	1	3.3	1	3.3
AA70	5	1992.8						0	11.4	0	11.4	
AA75	5	2102.8				0	9.8	0	10.8	0	10.8	
BB30	3	1017.6	0	0.4	0	1.4	0	5.1	0	5.9	0	5.9
BB35	3	1211.4	0	3.7	0	5.5	0	8.2	0	8.3	0	8.3
BB40	3	1503.2	1	−12.8	1	−12.4	1	−8.9	1	−6.8	1	−6.6
BB45	4	1386.5	0	1.5	0	3.6	0	9.6	0	10.8	0	10.9
BB50	4	1564.9	0	3.9	0	6.8	0	13.6	0	14.5	0	14.5
BB55	4	1801.4	1	−6.2		1	2.5	1	4.9	1	6.1	
BB60	6	2034.2	0	1.9	0	3.5	0	10.9	0	13.3	0	13.3
BB65	6	2212.3	0	2.0		0	15.7	0	15.9	0	15.9	
CC30	3	1054.0	0	0.0	0	0.0	0	1.6	0	2.9	0	3.3
CC35	3	1184.5	0	0.0	0	1.3	0	3.2	0	4.4	0	4.5
CC40	3	1279.5	0	0.0	0	0.0	0	1.7	0	4.2	0	4.8
CC50	4	1592.5										
AA′30	3	934.4	0	0.7	0	1.6	0	6.4	0	11.2	0	11.2
AA′35	3	1034.3	0	1.0	0	1.6	0	9.5	0	14.2	0	14.2
AA′40	3	1204.1	0	1.1	0	2.1	0	11.6	0	12.7	0	12.7
AA′45	3	1343.0	0	3.1	0	5.3	0	18.2	0	19.3	0	19.3
AA′50	3	1457.5	0	4.6	0	9.5	1	4.4	1	5.3	1	5.3
AA′55	4	1482.3	0	2.3		0	15.1	0	16.0	0	16.0	
AA′60	4	1605.2	0	3.0		0	16.5	0	17.5	0	17.5	
AA′65	4	1715.5								0	18.0	
BB′30	3	973.4	0	1.0	0	1.7	0	9.9	0	10.7	0	10.7
BB′35	3	1143.4	0	3.9	0	6.8	0	14.1	0	14.4	0	14.4
BB′40	3	1294.3	1	−3.3	1	−2.3	1	4.8	1	5.9	1	6.0
BB′45	3	1362.3	1	−3.8	1	−2.8	1	6.8	1	8.9	1	8.9
BB′50	4	1408.0	0	2.6		0	23.2	0	24.8	0	24.8	
BB′55	4	1565.8	0	6.1		1	13.1	1	15.8	1	17.5	

algorithm for the PDPTWL could solve five instances to optimality within the time limit for both the CC and DD group, where only three instances for the CC group and one instance for the DD group were solved to optimality by the algorithms for the PDPTWH-1 and the PDPTWH-2. For all groups, the average computation time is the least for the PDPTWL and the highest for the PDPTWH-2. Compared with the PDPTWL, the average computation time is 185.3% and 247.1% higher for the PDPTWH-1 and the PDPTWH-2, respectively. From this, we can conclude that allowing rehandling operations requires additional computation time for these instances. This can be explained by the more restrictive dominance

Table 6
Comparative computational results for the PDPTWL, PDPTWH-1, and the PDPTWH-2, the latter two with $\delta = 50$.

Group	PDPTWL		PDPTWH-1		PDPTWH-2	
	#	Sec.	#	ΔSec. (%)	#	ΔSec. (%)
AA	10	150.1	10	59.6	10	122.3
BB	10	286.4	10	24.8	10	57.5
CC	5	30.9	3	71.2	3	75.1
DD	5	154.4	1	4453.4	1	5079.3
Total	30	192.2	24	185.3	24	247.1

Table 7
Computational results for the PDPTWH-1.

Inst.	$\delta=0$			$\delta=0.5$			$\delta=1$			$\delta=5$			$\delta=10$			$\delta=50$		
	Sec.	$\underline{z}$	Z^*	Sec.	$\underline{z}$	z^*	Sec.	$\underline{z}$	z^*	Sec.	$\underline{z}$	Z^*	Sec.	$\underline{z}$	Z^*	Sec.	$\underline{z}$	z^*
AA30	1.8	30,969.4	30,969.4	16.9	30,986.3	30,988.4	2.1	30,990.8	30,990.8	6.9	31,064.0	31,067.4	2.0	31,129.5	31,129.5	1.9	31,129.5	31,129.5
AA35	26.1	31,084.4	31,089.0	5.9	31,090.4	31,090.4	6.5	31,114.2	31,114.2	24.3	31,222.6	31,230.8	16.4	31,285.2	31,294.1	16.7	31,285.2	31,294.1
AA40	96.5	41,235.1	41,241.7	148.7	41,248.1	41,254.6	188.9	41,261.1	41,269.3	25.3	41,324.0	41,325.8	4.0	41,345.7	41,345.7	4.2	41,349.3	41,349.3
AA45	181.4	41,399.5	41,412.2	154.1	41,416.3	41,422.3	1339.0	41,430.0	41,443.6	106.5	41,497.6	41,505.8	8.0	41,517.9	41,517.9	7.2	41,521.4	41,521.4
AA50	1843.1	41,507.3	41,531.6	138.2	41,532.6	41,536.8	1674.5	41,541.0	41,559.2	2579.1	41,621.8	41,634.9	16.1	41,643.7	41,643.7	18.9	41,643.7	41,643.7
AA55	450.1	41,653.5	41,667.1	326.7	41,687.2	41,696.9	383.7	41,756.7	41,765.4	130.5	46,767.5	51,700.9	481.4	46,800.8	51,743.2	24.5	46,803.8	51,743.2
AA60	1824.5	41,805.5	41,822.7	314.2	41,841.9	41847.9	482.4	41,912.3	41,921.1	407.9	46,967.7	51,894.8	999.8	46,998.1	51,947.9	1110.7	46,999.2	51,949.7
AA65	1229.8	42,001.8	42,011.6	242.5	42,876.8	51,887.1	233.9	43,753.7	51,899.7	263.8	47,157.4	52,021.6	429.6	47,172.0	52,077.2	260.0	47,172.0	52,077.4
AA70	686.0	42,672.8	51,992.8				7802.6	44,402.1	52,056.8				661.9	47,896.1	52,219.2	479.6	47,896.1	52,219.2
AA75	6589.4	45,576.3	52,102.8							780.0	51,151.9	52,308.2	478.3	51,605.8	52,330.1	471.0	51,607.2	52,330.1
BB30	2.5	31,017.6	31,017.6	2.2	31,021.4	31021.4	15.5	31,030.8	31,032.3	1.7	31,069.7	31,069.7	4.9	31076.3	31077.5	4.7	31,076.3	31,077.5
BB35	16.5	31,210.8	31,211.4	62.5	31,255.7	31,258.1	6.2	31,277.8	31,277.8	4.3	31,310.8	31,310.8	4.4	31,312.4	31,312.4	3.8	31,312.4	31,312.4
BB40	15.2	31,503.2	31,503.2	109.6	33,425.8	41,311.0	14.6	34,193.6	41,317.3	9.4	35,675.7	41,369.3	106.8	35,693.9	41,400.5	95.2	35,695.5	41,404.0
BB45	115.0	33,217.8	41,386.5	1179.7	34,910.3	41,413.2	853.0	35,891.5	41,441.4	1299.6	37,476.1	41,519.6	257.6	37,642.0	41,535.6	255.4	37,645.1	41,537.5
BB50	284.1	41,562.1	41,564.9	461.9	41,616.0	41,636.4	240.3	41,681.2	41,685.9	92.7	41,773.5	41,777.2	35.7	41,791.1	41,791.1	30.2	41,791.1	41,791.1
BB55	1000.1	41,786.7	41,801.4	4818.5	43,347.2	51,692.3				593.2	46,244.3	51,852.4	836.7	46,388.6	51,889.5	1023.5	46,391.4	51,911.7
BB60	316.3	62,026.0	62,034.2	755.1	62,061.0	62,072.5	1654.2	62,094.0	62,106.1	91.4	62,254.2	62,255.8	34.9	62,300.7	62,304.3	29.0	62,305.5	62,305.5
BB65	1291.2	62,199.2	62,212.3	4505.1	62,240.4	62,257.6				277.4	62,556.0	62,560.3	46.1	62,559.8	62,564.6	44.4	62,564.6	62,564.6
BB70										717.3	65,952.9	72,503.9	474.5	65,996.7	72,535.2	611.3	66,002.4	72,535.2
BB75										4,470.5	68,148.6	72,617.6	450.5	68,188.4	72,643.2	1477.1	68,197.0	72,656.7
CC30	7.8	23,248.6	31,054.0	10.7	23,282.4	31,054.0	9.1	23,283.3	31,054.0	47.9	23,297.5	31,071.3	24.2	23,318.6	31,085.0	25.8	23,318.9	31,088.6
CC35	30.6	24,551.0	31,184.5	21.3	24,731.6	31,184.5	37.2	24,742.2	31,199.8	102.0	24,762.5	31,222.1	82.9	24,777.3	31,236.5	51.1	24,777.3	31,237.5
CC40	587.6	25,412.5	31,279.5	203.9	25,751.0	31,279.5	272.0	25,835.2	31,279.5	217.9	26,006.0	31,300.9	96.5	26,024.2	31,333.5	81.8	26,024.6	31,340.3
CC50	10,174.6	34,104.3	41,592.5															
CC55										8325.2	36,886.7	41,747.3						
DD30	3346.3	18,259.1	20,993.6															
DD35										6331.4	21,904.9	31,113.5	8699.9	21,904.8	31,123.9	7030.5	21,904.8	31,127.8
AA'30	101.5	25,952.1	30,934.4	134.8	25,960.3	30,941.4	214.2	25,978.1	30,949.3	3.5	27,715.5	30,993.8	17.4	31,034.2	31,039.4	24.5	31,034.8	31,039.4
AA'35	128.9	27,735.2	31,034.3	149.3	27,761.7	31,045.0	16.5	27,786.9	31,050.4	64.7	31,128.5	31,133.0	585.2	31,168.9	31,180.8	616.7	31,168.9	31,180.8
AA'40	181.6	31,201.1	31,204.1	96.5	31,216.4	31,217.9	107.2	31,229.1	31,230.8	481.2	31,322.8	31,343.5	473.9	31,347.8	31,356.5	366.6	31,347.8	31,356.5
AA'45	474.3	31,327.7	31,343.0	295.3	31,380.9	31,387.3	685.7	31,399.4	31,414.8	69.2	31,587.1	31,587.1	80.1	31,602.3	31,602.3	71.5	31,602.3	31,602.3
AA'50	295.5	31,457.5	31,457.5	529.6	31,526.8	31,526.8	3560.2	31,584.6	31,647.6	8099.8	36,052.1	41,521.2	223.5	36,594.6	41,534.2	175.4	36,596.3	41,534.2
AA'55	215.0	41,479.3	41,482.3	4816.2	41,505.7	41,518.5			1775.5	41,699.5	41,706.6		57.8	41,720.1	41,720.1	74.4	41,720.1	41,720.1
AA'60	312.8	41,597.7	41,605.2						826.2	41,860.6	41,869.5	234.4	41,882.3	41,885.9		806.1	41,882.3	41,885.9
AA'65	354.1	41,715.5	41,715.5													4653.0	42,003.3	42,024.4
BB'30	4.0	30,973.4	30,973.4	2.7	30,983.3	30,983.3	12.5	30,989.6	30,990.1	2.1	31,069.7	31,069.7	7.1	31,076.3	31,077.5	7.2	31,076.3	31,077.5
BB'35	49.3	31,139.6	31,143.4	308.3	31,177.6	31,188.5	157.8	31,215.1	31,221.6	34.7	31,302.7	31,304.3	40.0	31,307.3	31,307.9	40.9	31,307.3	31,307.9
BB'40	24.3	31,294.3	31,294.3	372.7	31,903.8	41,251.6	123.6	33,062.8	41,264.1	2581.3	34,053.8	41,356.5	700.8	34,068.1	41,370.3	773.0	34,068.8	41,372.0
BB'45	137.5	31,362.3	31,362.3	2182.1	33,086.0	41,310.1	1173.7	34,165.2	41,323.6	256.6	36,540.1	41,455.2	3647.6	36,553.7	41,484.2	1614.3	36,555.0	41,484.2
BB'50	725.2	35,108.0	41,408.0	749.8	37,549.0	41,445.0				598.0	40,320.0	41,734.9	4348.2	40,392.6	41,757.3	2700.4	40,397.6	41,757.3
BB'55	2093.8	36,974.2	41,565.8	1460.7	40,215.0	41,661.5	5430.9	41,820.1	51,570.3	3476.9	44,205.3	51,776.4	679.1	44,336.9	51,813.8	639.0	44,343.4	51,840.5
BB'60						10,628.7	53,511.8	61,924.2		353.4	56,875.3	62,073.8	532.7	56,934.0	62,136.9	300.7	56,934.9	62,138.9
BB'65												801.4	58,176.5	62,328.5		304.8	58,177.6	62,329.0
BB'70																1415.0	60,774.9	62,547.4

Table 8
Computational results for the PDPTWH-2.

Inst.	$\delta=0$			$\delta=0.5$			$\delta=1$			$\delta=5$			$\delta=10$			$\delta=50$		
	Sec.	$\underline{z}$	Z^*	Sec.	$\underline{z}$	z^*	Sec.	$\underline{z}$	z^*	Sec.	$\underline{z}$	Z^*	Sec.	$\underline{z}$	Z^*	Sec.	$\underline{z}$	z^*
AA30	1.9	30,969.4	30,969.4	17.3	30,986.3	30,988.4	2.0	30,990.8	30,990.8	7.3	31,064.0	31,067.4	2.1	31,129.5	31,129.5	1.9	31,129.5	31,129.5
AA35	25.1	31,084.4	31,089.0	6.5	31,090.4	31,090.4	6.1	31,114.2	31,114.2	58.3	31,222.6	31,230.8	17.4	31,285.2	31,294.1	17.7	31,285.2	31,294.1
AA40	79.8	41,235.1	41,241.7	153.5	41,248.1	41,254.6	205.2	41,261.1	41,269.3	27.3	41,324.0	41,325.8	4.2	41,345.7	41,345.7	4.4	41,349.3	41,349.3
AA45	195.0	41,399.5	41,412.2	140.7	41,416.3	41,422.3	1166.3	41,430.0	41,443.6	106.8	41,497.6	41,505.8	8.5	41,517.9	41,517.9	7.7	41,521.4	41,521.4
AA50	929.7	41,507.3	41,531.6	220.1	41,532.6	41,536.8	2743.9	41,541.0	41,559.2	1149.6	41,621.8	41,634.9	17.3	41,643.7	41,643.7	20.6	41,643.7	41,643.7
AA55	418.4	41,653.5	41,667.1	421.8	41,687.2	41,696.9	330.4	41,756.7	41,765.4	157.9	46,767.5	51,700.9	494.5	46,800.8	51,743.2	26.5	46,803.8	51,743.2
AA60	1886.0	41,805.5	41,822.7	1221.0	41,841.9	41,847.9	460.7	41,912.3	41,921.1	1116.5	46,967.7	51,894.8	1062.9	46,998.1	51,947.9	1312.0	46,999.2	51,949.7
AA65	564.7	42,001.8	42,011.6	206.6	42,876.8	51,887.1	255.4	43,753.7	51,899.7	249.8	47,157.4	52,021.6	423.5	47,172.0	52,077.2	346.8	47,172.0	52,077.4
AA70	491.3	42,672.8	51,992.8										842.9	47,896.1	52,219.2	833.0	47,896.1	52,219.2
AA75	5559.1	45,576.3	52,102.8							1599.1	51,151.9	52,308.2	545.6	51,605.8	52,330.1	766.4	51,607.2	52,330.1
BB30	2.2	31,017.6	31,017.6	1.9	31,021.4	31,021.4	16.2	31,030.8	31,032.3	1.5	31,069.7	31,069.7	5.2	31,076.3	31,077.5	4.9	31,076.3	31,077.5
BB35	16.7	31,210.8	31,211.4	36.3	31,255.1	31,256.7	7.4	31,277.8	31,277.8	5.1	31,310.8	31,310.8	4.8	31,312.4	31,312.4	4.1	31,312.4	31,312.4
BB40	15.3	31,503.2	31,503.2	141.7	33,296.2	41,311.0	16.7	34,193.0	41,317.3	45.5	35,675.7	41,369.3	113.6	35,693.9	41,400.5	98.5	35,695.5	41,404.0
BB45	115.0	33,217.8	41,386.5	292.2	34,891.6	41,407.2	1786.6	35,821.0	41,436.5	3074.8	37,476.1	41,519.6	266.7	37,642.0	41,535.6	262.3	37,645.1	41,537.5
BB50	303.4	41,562.1	41,564.9	288.8	41,613.1	41,625.6	98.0	41,671.6	41,671.6	111.8	41,773.5	41,777.2	42.7	41,791.1	41,791.1	36.1	41,791.1	41,791.1
BB55	1000.7	41,786.7	41,801.4	7898.2	43,223.6	51,689.5				366.4	46,238.7	51,846.6	892.0	46,388.6	51,889.5	1205.0	46,391.4	51,911.7
BB60	262.9	62,026.0	62,034.2	324.2	62,060.4	62,072.5	4778.4	62,094.0	62,106.1	350.7	62,254.2	62,255.8	40.4	62,300.7	62,304.3	46.7	62,305.5	62,305.5
BB65	3040.4	62,199.2	62,212.3	4,663.3	62,239.6	62,257.6				275.0	62,556.0	62,560.3	60.7	62,559.8	62,564.6	78.1	62,564.6	62,564.6
BB70										970.9	65,952.9	72,503.9	529.9	65,996.7	72,535.2	831.4	66,002.4	72,535.2
BB75										3326.9	68,148.6	72,617.6	529.9	68,188.4	72,643.2	1944.7	68,197.0	72,656.7
CC30	8.0	23,248.6	31,054.0	10.9	23,282.4	31,054.0	9.3	23,283.3	31,054.0	49.1	23,297.5	31,071.3	24.8	23,318.6	31,085.0	26.4	23,318.9	31,088.6
CC35	31.1	24,551.0	31,184.5	21.7	24,731.6	31,184.5	37.9	24,742.2	31,199.8	104.2	24,762.5	31,222.1	84.6	24,777.3	31,236.5	52.2	24,777.3	31,237.5
CC40	590.8	25,412.5	31,279.5	839.1	25,751.0	31,279.5	276.1	25,835.2	31,279.5	223.6	26,006.0	31,300.9	98.8	26,024.2	31,333.5	83.6	26,024.6	31,340.3
CC50	10,091.9	34,104.3	41,592.5															
CC55							5808.9	36,480.3	41,708.4									
DD30																		
DD35										7012.6	21,904.9	31,113.5	9384.5	21,904.8	31,123.9	7996.8	21,904.8	31,127.8
AA'30	112.3	25,952.1	30,934.4	176.4	25,960.3	30,941.4	288.1	25,977.5	30,949.3	4.1	27,715.5	30,993.8	29.4	31,034.2	31,039.4	27.9	31,034.8	31,039.4
AA'35	176.8	27,735.2	31,034.3	153.4	27,761.7	31,045.0	19.0	27,785.7	31,050.4	76.8	31,128.5	31,133.0	559.9	31,168.9	31,180.8	548.6	31,168.9	31,180.8
AA'40	93.1	31,201.1	31,204.1	108.3	31,216.4	31,217.9	107.5	31,228.4	31,229.6	1062.9	31,322.8	31,343.5	575.6	31,347.8	31,356.5	439.6	31,347.8	31,356.5
AA'45	7116.9	31,327.7	31,343.0	280.5	31,380.9	31,384.6	952.9	31,397.0	31,413.6	88.7	31,587.1	31,587.1	97.5	31,602.3	31,602.3	86.5	31,602.3	31,602.3
AA'50	323.4	31,457.5	31,457.5	492.5	31,524.6	31,524.6	1604.1	31,583.5	31,596.1	2922.7	36,052.1	41,521.2	265.7	36,594.6	41,534.2	208.6	36,596.3	41,534.2
AA'55	177.2	41,479.3	41,482.3	943.5	41,505.7	41,515.8				1706.3	41,699.5	41,706.6	68.8	41,720.1	41,720.1	90.9	41,720.1	41,720.1
AA'60	334.4	41,597.7	41,605.2	2447.1	41,637.5	41,653.7				1039.0	41,860.6	41,869.5	278.9	41,882.3	41,885.9	1002.8	41,882.3	41,885.9
AA'65	384.2	41,715.5	41,715.5													7143.7	42,003.3	42,024.4
BB'30	3.6	30,973.4	30,973.4	2.7	30,983.3	30,983.3	13.8	30,989.6	30,990.1	2.0	31,069.7	31,069.7	7.8	31,076.3	31,077.5	7.9	31,076.3	31,077.5
BB'35	47.9	31,139.6	31,143.4	233.6	31,175.8	31,188.1	372.5	31,213.8	31,221.6	40.9	31,302.7	31,304.3	42.1	31,307.3	31,307.9	44.1	31,307.3	31,307.9
BB'40	24.7	31,294.3	31,294.3	304.9	31,843.3	41,251.6	130.6	33,052.8	41,264.1	2195.3	34,053.8	41,356.5	1307.4	34,068.1	41,370.3	873.3	34,068.8	41,372.0
BB'45	107.7	31,362.3	31,362.3	1517.4	32,978.9	41,310.1	425.0	34,029.8	41,323.6	347.2	36,540.1	41,455.2	3986.7	36,553.7	41,484.2	1820.5	36,555.0	41,484.2
BB'50	607.1	35,108.0	41,408.0	906.7	37,348.7	41,445.0				671.8	40,320.0	41,734.9	6632.7	40,392.6	41,757.3	3857.6	40,397.6	41,757.3
BB'55	1159.1	36,974.2	41,565.8	2251.8	39,979.3	41,661.5				2358.8	44,205.3	51,771.5	1070.9	44,336.9	51,813.8	902.3	44,343.4	51,840.5
BB'60										458.7	56,873.4	62,068.6	620.6	56,934.0	62,136.9	369.3	56,934.9	62,138.9
BB'65													994.1	58,176.5	62,328.5	338.5	58,177.6	62,329.0
BB'70										5783.2	60,437.6	62,478.9				2110.9	60,774.9	62,547.4

criterion for the PDPTWH-1 compared with the PDPTWL, which implies that the PDPTWH-1 is harder to solve. When comparing the PDPTWH-2 with the PDPTWL, we see that there are more label extensions in the algorithm for the PDPTWH-2, which makes the PDPTWH-2 harder to solve. Comparing the average computation time for the optimal solutions of the PDPTWH-1 and the PDPTWH-2, we observe for all instance groups that the average time to solve the instances to optimality is larger for the PDPTWH-2 compared with the PDPTWH-1. This implies that allowing the more general rehandling policy 2 requires additional time. The major differences in the algorithms for the PDPTWH-1 and the PDPTWH-2 are the dominance criteria and the label extension functions. The dominance criterion for the PDPTWH-1 is the most restrictive. This implies that relatively more labels are eliminated by the dominance criterion for the PDPTWH-2 compared with the PDPTWH-1. However, whereas in the algorithm for the PDPTWH-1 only one new label is created when extending a label along an arc, multiple labels are created when extending a label along an arc in the algorithm for the PDPTWH-2. This results in a large number of labels that are created for the PDPTWH-2. The shorter average computation time for the PDPTWH-1 in comparison with the PDPTWH-2 for $\delta = 50$ suggests that for these instances the relatively larger number of labels eliminated in the PDPTWH-2 does not counterbalance the large number of extra labels created.

5. Conclusions

We have introduced the pickup and delivery problem with time windows and handling operations, which arises in the transportation of heavy, dangerous or large items in a less-than-truckload setting. For this problem, we have defined and analyzed two different rehandling policies. The first rehandling policy only allows compulsory rehandling. The second policy is a generalization of the first one, where compulsory rehandling must be done and preventive rehandling is allowed. For both policies, we have developed a specific branch-price-and-cut algorithm, where the pricing problems correspond to elementary shortest path problems with pickups and deliveries, time windows, a capacity constraint, and rehandling operations. To solve the pricing problems, we have developed labeling algorithms that consider the relative positions of items in a vehicle and thereby break symmetry between orders of items. Non-straightforward dominance criteria were proposed for both policies. The labeling algorithm for policy 2 extends each label by multiple labels to account for each feasible combination of rehandled items. Both branch-price-and-cut algorithms are able to solve instances with up to 75 requests to optimality. For the instances under study, the travel costs can be reduced by up to 3.2% by allowing rehandling policy 2 instead of the more restrictive rehandling policy 1. However, more instances were solved by the algorithm under policy 1 and the average computation time for the algorithm under policy 2 is larger than for the algorithm under policy 1, which can be explained by the generation of a large number of labels in the labeling algorithm for policy 2. In conclusion, even though policy 2 allows more flexibility in the rehandling operations, it does not always results in a larger cost reduction compared with policy 1 for the instances studied in this paper, and the problem becomes harder to solve. For the instances under study, the PDPTWH under policy 1 already provides interesting insights. Compared with the PDPTWL it is more flexible since it allows rehandling operations. For other instances with larger vehicle capacities and different time windows, investigating additional handling policies could be interesting. The algorithms developed in this paper could be adapted to other versions of the PDPTW, such as the PDPTWMS where rehandling operations could be allowed, or to a version of the PDPTWH that would minimize the total route duration as a secondary objective. Another area of future research is the development of heuristics for the PDPTWH.

Acknowledgments

This project was funded by the Dutch Institute for Advanced Logistics (Dinalog) and by the Canadian Natural Sciences and Engineering Research Council under grants 157935-2012 and 2015-06189. This support is gratefully acknowledged. The authors are thankful to François Lessard for his valuable advice. Thanks are due to the referees for their valuable comments.

Appendix

Tables 7 and 8 report detailed computational results for the PDPTWH-1 and the PDPTWH-2, respectively. Each table contains *Inst*, the name of the instance defined as its group and the number of requests, and for each value of δ, *Sec.*, the total computation time in seconds, $\underline{z}$, the value of the lower bound obtained at the root node before adding cuts, and z^*, the value of the optimal solution. For all instances that were not solved within the prescribed time limit, we do not report a lower bound value.

References

[1] Battarra M, Erdoğan G, Laporte G, Vigo D. The traveling salesman problem with pickups, deliveries, and handling costs. Transp Sci 2010;44(3):383–99.
[2] Berbeglia G, Cordeau J-F, Gribkovskaia I, Laporte G. Static pickup and delivery problems: a classification scheme and survey. TOP 2007;15(1):1–31.
[3] Cherkesly M. Le problème de tournées de véhicules avec cueillettes, livraisons, fenêtes de temps en contraintes de manutention [Ph.D. thesis]. École Polytechnique de Montréal; 2015.
[4] Cherkesly M, Desaulniers G, Irnich S, Laporte G. Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and multiple stacks. Eur J Oper Res 2016;250(3):782–93.
[5] Cherkesly M, Desaulniers G, Laporte G. Branch-price-and-cut algorithms for the pickup and delivery problem with time windows and last-in-first-out loading. Transp Sci 2015;49(4):752–66.
[6] Cherkesly M, Desaulniers G, Laporte G. A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. Comput Oper Res 2015;62:23–35.
[7] Christofides N, Mingozzi A, Toth P. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. Math Progr 1981;20(1):255–82.
[8] Desaulniers G, Desrosiers J, Spoorendonk S. Cutting planes for branch-and-price algorithms. Networks 2011;58(4):301–10.
[9] Desaulniers G, Lessard F, Hadjar A. Tabu search, partial elementarity, and generalized *k*-path inequalities for the vehicle routing problem with time windows. Transp Sci 2008;42(3):387–404.
[10] Erdoğan G, Battarra M, Laporte G, Vigo D. Metaheuristics for the traveling salesman problem with pickups, deliveries and handling costs. Comput Oper Res 2012;39(5):1074–86.
[11] Jepsen M, Petersen B, Spoorendonk S, Pisinger D. Subset-row inequalities applied to the vehicle-routing problem with time windows. Oper Res 2008;56(2):497–511.
[12] Kohl N, Desrosiers J, Madsen OBG, Solomon MM, Soumis F. 2-path cuts for the vehicle routing problem with time windows. Transp Sci 1999;33(1):101–16.
[13] Naddef D, Rinaldi G. Branch-and-cut algorithms for the capacitated VRP. In: Toth P, Vigo D, editors. The vehicle routing problem. SIAM monographs on discrete mathematics and applications, Philadelphia: Society for Industrial and Applied Mathematics; 2002. p. 53–84, [Chaper 3].
[14] Parragh S, Doerner KF, Hartl RF. A survey on pickup and delivery problems part I: transportation between customers and depot. J Betr 2008;58(1):21–51.

[15] Parragh S, Doerner KF, Hartl RF. A survey on pickup and delivery problems, part II: transportation between pickup and delivery locations. J Betr 2008;58 (2):81–117.
[16] Ropke S, Cordeau J-F. Branch-and-cut-and-price for the pickup and delivery problem with time windows. Transp Sci 2009;43(3):267–86.
[17] Ropke S, Cordeau J-F, Laporte G. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. Networks 2007;49(4):258–72.
[18] Savelsbergh MWP, Sol M. The general pickup and delivery problem. Transp Sci 1995;29(1):17–29.
[19] Veenstra M, Roodbergen KJ, Vis IFA, Coelho LC. The pickup and delivery traveling salesman problem with handling costs. Eur J Oper Res 2016.