



# The pickup and delivery problem with time windows and scheduled lines

Veaceslav Ghilas, Emrah Demir & Tom Van Woensel

To cite this article: Veaceslav Ghilas, Emrah Demir & Tom Van Woensel (2016) The pickup and delivery problem with time windows and scheduled lines, INFOR: Information Systems and Operational Research, 54:2, 147-167, DOI: [10.1080/03155986.2016.1166793](https://doi.org/10.1080/03155986.2016.1166793)

To link to this article: <https://doi.org/10.1080/03155986.2016.1166793>



© 2016 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 04 Apr 2016.



Submit your article to this journal [↗](#)



Article views: 3506



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 3 View citing articles [↗](#)

# The pickup and delivery problem with time windows and scheduled lines

Veaceslav Ghilas, Emrah Demir and Tom Van Woensel

Department of Industrial Engineering and Innovation Sciences, Operations, Planning, Accounting and Control (OPAC), Eindhoven University of Technology, Eindhoven, The Netherlands

## ABSTRACT

Integrating freight flows with scheduled public transportation services creates attractive business opportunities as the same transportation needs can be met with fewer operating costs. The pickup and delivery problem with time windows and scheduled lines (PDPTW-SL) aims at routing a given set of vehicles to transport freight requests from their origins to their corresponding destinations, where the requests can use scheduled passenger transportation services as a part of their journeys. We describe the PDPTW-SL as an arc-based mixed-integer program. Computational results on a set of small-size instances provide a clear understanding of the benefits of using scheduled line services as a part of freight's journey.

## ARTICLE HISTORY

Received 11 July 2015  
Accepted 6 March 2016

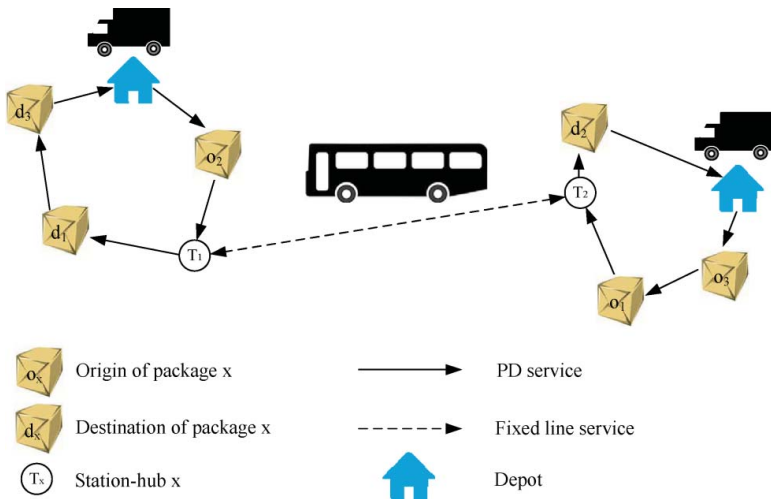
## KEYWORDS

Freight transportation; multi-modal transportation; routing and scheduling; scheduled lines

## 1. Introduction

A successful integration of freight and public transportation creates a seamless movement for both people and goods (i.e. small parcels). Actual integration is already being observed in long-haul freight transportation (Levin et al. 2012). For example, Norwegian Hurtigruten carries freight and people at the same time (Hurtigruten 2013). In short-haul transportation, however, people and freight rarely share transportation modes although they largely use the same infrastructure (Lindholm & Behrends 2012). Knowing this reality, potential efficiency gains for an integrated approach can be further investigated. To the best of our knowledge, integrated transport solutions to short-haul transportation have not been sufficiently taken into consideration in the literature.

This paper focuses on opportunities to make use of available public transportation, which operates according to predetermined routes and schedules, as a part of freight journey. We assume that each considered public transport vehicle has a finite carrying capacity for freight requests apart from available spaces destined for passengers. Therefore, transferring freight requests to available scheduled lines (SLs) could be beneficial for the whole transportation system. We consider two transport options: direct and indirect shipments. The first one implies that origin and destination points of a request are visited by using only one pickup and delivery (PD) vehicle. On the other hand, the second type of



**Figure 1.** An illustration of the integrated transportation network.

shipment implies that a request is picked up by a PD vehicle and transferred to a transfer node, which is assumed to be located nearby. From there, the request continues its journey on SLs. Afterwards, the request is picked up again by another PD vehicle to be delivered to its destination point. In this paper, we denote this specific transportation problem as the pickup and delivery problem with time windows and scheduled lines (PDPTW-SL). A schematic overview of the example network is shown in Figure 1.

Figure 1 presents three requests that have their PD locations close to two different transfer nodes. It would make sense to use a SL that connects these two transfer nodes instead of using one PD vehicle as in the pickup and delivery problem (PDP). Hence, travel time savings of PD vehicles and operational cost reduction can be expected with the proposed integrated system.

It is noteworthy that a small-sized freight (i.e. packages) is considered and the consolidation of the requests is done on both modes of transportation. In other words, a PD vehicle can perform several pickups before it delivers the requests to their delivery nodes, hence, carrying larger quantities. Similarly, SLs facilitate for consolidation in the start node, thus multiple freight requests can be transported on the SL vehicles at a time (as long as the capacity constraint is not violated). Hence, freight consolidation leads to higher efficiency of the whole transportation system.

The contributions of the paper are the following: (1) we introduce a mixed-integer program for the PDPTW-SL, (2) we propose some effective valid inequalities for the given formulation and (3) we analyse the benefits of using SLs in a PD environment by comparing the solutions of the PDPTW with the solution of the PDPTW-SL. The remainder of the paper is structured as follows. Section 2 provides a brief review of the related works. Section 3 introduces a mixed-integer programming (MIP) formulation for the PDPTW-SL. Section 4 reports the results obtained from extensive computational experiments. Section 5 provides conclusions and future research directions.

## 2. Literature review

This section reviews the literature addressing the related problems of the PDPTW-SL. To the best of our knowledge, there is only limited literature on transfer options to SL services. Thus, our paper is one of the first attempts to investigate integrated short-haul freight transportation from an operational planning perspective.

In the context of PDP, we refer to Berbeglia et al. (2007) for a literature survey on solution methodologies. Cordeau and Laporte (2007) provide an overview of mathematical models and solution methodologies for single- as well as multi-vehicle dial-a-ride problem (DARP), which is an extension of the PDP. In a related study by Parragh (2010), the author investigates heterogeneity in vehicles and passengers along with waiting times of passengers. The author proposes a Branch & Cut (B&C) and a variable neighbourhood search (VNS) algorithm to solve the DARP. Parragh et al. (2012) extend the heterogeneous DARP to consider driver-related constraints and propose a hybrid column generation and VNS heuristic algorithm.

One of the PDP extensions considering transfer opportunity between PD vehicles at transfer nodes is coined as PDP with transfers (PDP-T). In the PDP-T, a request can be picked up by one vehicle and delivered by another. A relevant study by Shang and Cuff (1996) is one of the earliest works on the PDP-T. The authors propose an insertion heuristic algorithm which leads to a cost reduction of up to 37% compared to the manual plans. In the work of Cortes et al. (2010), the authors propose an exact decomposition method, namely B&C, to solve the PDP-T. The proposed algorithm provides savings of up to 90% in terms of CPU time compared to traditional Branch & Bound (B&B) algorithm. In another work, Masson et al. (2012) introduce an adaptive large neighbourhood search (ALNS) algorithm to solve the PDP-T. Their results indicate savings of up to 9% in terms of travel time, obtained by using intermediate transfers. In addition, Masson et al. (2014) propose an ALNS for the DARP with transfers (DARPT). The authors show that using transfers between vehicles may lead to significant savings in terms of operating costs (i.e. up to 8%). In addition, Petersen and Røpke (2011) investigate the PDP with cross-docking opportunities (PDPCD). The authors propose a large neighbourhood search heuristic for the PDPCD. Their results suggest that improvements of up to 5% can be achieved due to cross-docking operations.

One of the first authors that conceptually introduced the idea of transporting freight using public transport was Nash (1982). Later, Liaw et al. (1996) investigate the opportunity to integrate SL services with the DARP. The authors formulate the problem where transshipment to public scheduled transportation is allowed for passengers and wheel-chaired persons and propose two types of heuristics (i.e. online and offline). Their computational results show that improvements of up to 9% in terms of the number of serviced requests can be obtained using static planning and up to 7% with the help of online decision support. In another work, Aldaihani and Dessouky (2003) consider an integrated DARP with public transport and propose a two-stage heuristic algorithm (i.e. insertion and tabu search). Computational results indicate that solutions generated by the proposed algorithm lead to cost savings of 16.6%. In a related work of Häll et al. (2009), the authors introduce a MIP formulation to solve an integrated DARP. The authors consider transfers to fixed lines without modelling the schedule of the public transportation.

**Table 1.** An overview of the related literature.

Research	Multi-depots	Hetero-geneous fleet	Transfers among PD vehicles	Transfers to fixed lines		Dyna-mics	Solution approach
				No sched. <sup>a</sup>	Sched. <sup>b</sup>		
Shang and Cuff (1996)	—	—	✓	—	—	—	Heuristic
Cortes et al. (2010)	✓	✓	✓	—	—	—	B&C
Masson et al. (2012)	—	—	✓	—	—	—	Metaheuristic
Masson et al. (2014)	✓	—	✓	—	—	—	Metaheuristic
Liaw et al. (1996)	—	—	—	—	✓	✓	Heuristic
Häll et al. (2009)	—	—	—	✓	—	—	B&B
Trentini et al. (2012)	—	—	—	—	✓	—	Metaheuristic
This paper	✓	✓	—	—	✓	—	B&B

<sup>a</sup>Time schedules of the fixed lines are not considered.

<sup>b</sup>Time schedules of the fixed lines are considered.

‘—’: the aspect is not considered.

‘✓’: the aspect is considered.

Trentini and Malhene (2010) study an overview of solutions for combining freight and passenger transportation that are used in practice. The authors divide transport solutions into three categories: shared road capacities (multi-use lanes, night deliveries, etc.), shared public transport services (buses, subway, etc.) and, finally, shared consolidation facilities (delivery bays, lockers, etc.). In a related study, Trentini et al. (2012) investigate a two-echelon vehicle routing problem with transshipment in the context of public and freight integrated transportation system. The authors propose to use public transportation to ship the goods from a central distribution centre to predefined stations (e.g. transshipment points). From there, a number of tricycles are used to deliver products to their final destinations. The authors propose ALNS along with a MIP formulation. Based on their results, the integrated solution proved to be more efficient in terms of pollution compared to traditional routing.

Table 1 provides an overview of the most relevant literature. We note that all papers in the table consider time window (TW) aspect.

As seen in Table 1, the literature on SL services is limited. Therefore, we aim at creating a building block for further research in this area, considering the extra complexity due to integrating SLs with current road transport flows.

### 3. Mathematical formulations for the PDPTW-SL

In this section, we introduce a MIP formulation for the PDPTW-SL. All information (requests, demands, travel times and TWs) is assumed to be known in advance and an initial plan for the whole planning horizon (e.g. one day) is generated. More specifically, the plan is generated before operating day commences by considering only the requests known at the moment of planning (i.e. static planning). A solution to our model is a routing plan for the PD vehicles such that each request is served. Additionally, a time schedule for the PD vehicles and to-be-served requests is produced. The aim is to minimize the total operating costs. These include (1) operating (routing) cost of the PD vehicles and (2) the total transfer cost which is proportional to the number of shipped requests on the SLs.

### 3.1. Definitions and assumptions

We define the PDPTW-SL on a digraph  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ , where  $\mathcal{N}$  represents the set of nodes and  $\mathcal{A}$  represents the set of arcs. All other definitions and assumptions are described below.

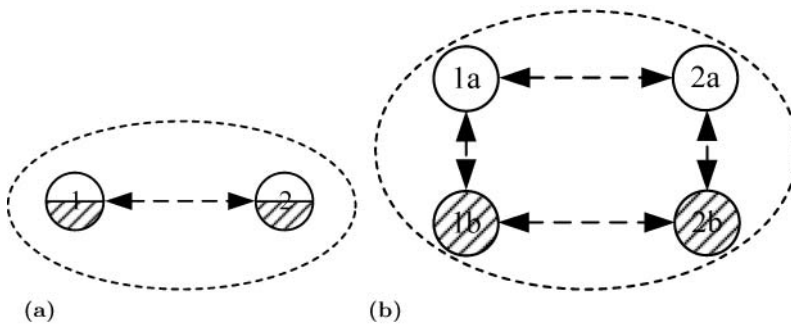
**Requests.** Every request  $r$  involves a pickup node  $r$  and a delivery node  $r + n$ , with  $n$  being the total number of requests that need to be services. In addition, any request  $r$  has two required TWs: one for the pickup node  $([l_r, u_r])$  and one for the delivery node  $([l_{r+n}, u_{r+n}])$ . The set of all requests is provided as  $\mathcal{P}$ . Note that we refer to a request by referring to its corresponding pickup node. Finally, demand  $d_r$  is known for each request.

**PD vehicles.** The set  $\mathcal{V}$  includes all available PD vehicles. Furthermore,  $Q_v$  corresponds to the carrying capacity of PD vehicle  $v$ . The origin and destination of every PD vehicle is given as node  $o_v$  with its TW  $[l_{o_v}, u_{o_v}]$ .

**Travelling and service times.** Travelling times, along with service times are assumed to be static and deterministic. Constant  $Y_{ij}$  represents the travel time from node  $i$  to node  $j$ . In addition, parameter  $s_i$  corresponds to the service time at node  $i \in \mathcal{N}$ .

**Scheduled lines.** Set  $\mathcal{S}$  contains all physical transfer nodes (i.e. end-stations of the SLs). Therefore,  $\mathcal{E}$  represents the set of all physical SLs. Each SL  $(i, j)$  connects the end-stations such that  $i, j \in \mathcal{S}$ . From the modelling perspective, note that arcs  $(i, j)$  and  $(j, i)$ ,  $i, j \in \mathcal{S}$ , represent two SLs. Each fixed line  $(i, j)$  has a set of indices  $\mathcal{K}^{ij}$  for the scheduled departures from node  $i$ , such that  $p_{ij}^w$  represents the departure time,  $\forall (i, j) \in \mathcal{E}$ ,  $w \in \mathcal{K}^{ij}$ . Note that each SL may have different frequency than other lines, thus the size of  $\mathcal{K}^{ij}$  may differ. Furthermore, it is assumed that SL vehicles are designed to carry a limited amount of packages, thus implying a finite carrying capacity  $k_{ij}$ ,  $\forall (i, j) \in \mathcal{E}$ . It is assumed that package carrying capacity is not influenced by the passenger demand.

Each physical transfer node is assumed to be a consolidation location for the packages (e.g. DHL-Packstation 2013). Namely, a secured area for temporary storage of the packages is available. The packages can be stored until their departure time (either by fixed line or PD vehicle) at these stations. Furthermore, as multiple freight carriers may be using SL services, we assume that each carrier is assigned a part of the storage at the transfer nodes and SL vehicles capacity (e.g. contract-based agreement). Hence, the considered capacity is not affected by the demand of another actors involved. In addition, it is also assumed that



**Figure 2.** An example of a replicated scheduled line. (a) Physical scheduled line and (b) virtual scheduled line.

cost  $\eta_{ij}$  per each unit of parcels shipped on the SL  $(i, j)$  includes transportation, handling (transshipment) and storage costs. Handling of the packages is assumed to be the responsibility of the SL service provider (e.g. a driver of the SL vehicle). Additionally, as the focus of the present problem is on the operational aspects, we disregard all investments needed, e.g. re-design of the SL vehicles and storage facilities at transfer nodes.

For modelling purposes, each physical SL  $(i, j) \in \mathcal{E}$  (e.g. arcs  $(1, 2)$  and  $(2, 1)$  in the example shown in Figure 2(a)) gets  $n$  replications as in Häll et al. (2009). This is done to consider multiple visits and waiting times of the PD vehicles at the transfer nodes. In addition, each replication is assigned to one request, and only that specific request can be shipped on the assigned SL (refer to Figure 2(b)). Therefore,  $\mathcal{F}$  contains all replicated SLs (e.g. arcs  $(1a, 2a)$ ,  $(1b, 2b)$ ,  $(2a, 1a)$  and  $(2b, 1b)$  in Figure 2(b)).

As a consequence, every physical transfer node in  $\mathcal{S}$  (e.g. nodes 1 and 2 in Figure 2(a)) gets  $n$  replications, thus obtaining the set of replicated transfer nodes as  $\mathcal{T}$  (e.g.  $\mathcal{T} \equiv \{1a, 1b, 2a, 2b\}$  in Figure 2(b)). For modelling purposes, let  $\psi^t, \forall t \in \mathcal{T}$  show the physical

**Table 2.** Parameters.

Notation	Definition
$\delta$	Number of depots
$Y_{ij}$	Travelling time from node $i$ to node $j$
$s_i$	Service time at node $i$
$k_{ij}$	Package carrying capacity on the fixed line $(i, j)$
$[l_i, u_i]$	Time window at node $i$
$\rho_{ij}^w$	Departure time from node $i$ , on the fixed line $(i, j)$ , indexed by $w$
$\tau$	The amount of replicated transfer nodes (i.e. $ \mathcal{T} $ )
$f_i^r$	$\begin{cases} 1 & \text{if node } i \text{ is the origin node of request } r, \\ 0 & \text{if node } i \text{ is an intermediate node, } \forall i \in \mathcal{N}_2 \setminus \{r; r+n\} \text{ (see Table 3)} \\ -1 & \text{if node } i \text{ is the destination of } r. \end{cases}$
$\phi_v$	The routing cost of one time unit for vehicle $v$
$\eta_{ij}$	The cost of shipping one unit of parcel on the SL $(i, j)$
$Q_v$	Freight carrying capacity of vehicle $v$

**Table 3.** Sets.

Notation	Definition
$\mathcal{O}$	Set of depots, $\mathcal{O} \equiv [1, \dots, \delta]$ (i.e. $\mathcal{O}_v \in \mathcal{O}, \forall v \in \mathcal{V}$ )
$\mathcal{P}$	Set of requests (pickup nodes), $\mathcal{P} \equiv [\delta + 1, \dots, \delta + n]$
$\mathcal{D}$	Set of delivery nodes, $\mathcal{D} \equiv [\delta + n + 1, \dots, \delta + 2n]$
$\mathcal{T}$	Set of replicated transfer nodes, $\mathcal{T} \equiv [\delta + 2n + 1, \dots, \delta + 2n + \tau]$ (e.g. nodes $1a, 1b, 2a$ and $2b$ in Figure 2(b))
$\mathcal{T}^t$	Set of replicated transfer nodes that represent the same physical transfer node as $t$ (e.g. in Figure 2(b), $\mathcal{T}^{2a} = \{2b\}$ , $\mathcal{T}^{1a} = \{1b\}$ , etc.)
$\mathcal{N}$	Set of nodes in $\mathcal{G}$ ; $\mathcal{P} \cup \mathcal{D} \cup \mathcal{O} \cup \mathcal{T} \equiv \mathcal{N}$
$\mathcal{N}_1$	$\mathcal{P} \cup \mathcal{D}$
$\mathcal{N}_2$	$\mathcal{P} \cup \mathcal{D} \cup \mathcal{T}$
$\mathcal{V}$	Set of PD vehicles
$\mathcal{E}$	Set of physical fixed lines which is defined as $(i, j)$ with associated $\mathcal{K}^{ij}$ and $k_{ij}$
$\mathcal{K}^{ij}$	Set of indices for the departure times from the physical transfer node $i$ on fixed line $(i, j) \in \mathcal{E}$
$\mathcal{F}$	Set of replicated fixed lines which is defined as $(i, j)$ with associated $\mathcal{K}^{\psi^i \psi^j}$
$\mathcal{F}^r$	Set of replicated fixed lines associated with request $r$ (e.g. in Figure 2 $\mathcal{F}^a = \{(1a, 2a), (2a, 1a)\}$ )
$\mathcal{F}^t$	Set of replicated fixed lines connected to the replicated transfer node $t$ (e.g. in Figure 2, $\mathcal{F}^{1a} = \{(1a, 2a), (2a, 1a)\}$ )
$\mathcal{F}^{ij}$	Set of replicated fixed lines associated with a physical fixed line $(i, j) \in \mathcal{E}$ (e.g. in Figure 2, $\mathcal{F}^{1,2} = \{(1a, 2a), (1b, 2b)\}$ , $\mathcal{F}^{2,1} = \{(2a, 1a), (2b, 1b)\}$ )
$\mathcal{A}$	Set of arcs in $\mathcal{G}$ defined by $\mathcal{N} \times \mathcal{N}$ , (note that $\mathcal{A} \equiv \mathcal{F} \cup \{(i, j)   i \in \mathcal{O}, j \in \mathcal{N}_2\} \cup \{(i, j)   i \in \mathcal{N}_2, j \in \mathcal{O}\}$ )
$\mathcal{A}^1$	$= \mathcal{N}_2 \times \mathcal{N}_2 \setminus \mathcal{F}$

**Table 4.** Decision variables.

Notation	Definition
$x_{ij}^v$	A binary variable with value 1 if arc $(i, j)$ is traversed by PD vehicle $v$ , 0 otherwise, $\forall (i, j) \in \mathcal{A}, v \in \mathcal{V}$
$\alpha_v$	A continuous variable that indicates the time vehicle $v$ returns to its depot, $\forall v \in \mathcal{V}$
$\beta_i$	A continuous variable that indicates the departure time of a PD vehicle from node $i$ , $\forall i \in \mathcal{N}$
$y_{ij}^r$	A binary variable with value 1 if arc $(i, j)$ is traversed by request $r$ , 0 otherwise, $\forall i, j \in \mathcal{N}_2, r \in \mathcal{P}$
$\gamma_i^r$	A continuous variable that indicates the departure time of request $r$ from node $i$ , $\forall i \in \mathcal{N}_2, r \in \mathcal{P}$
$q_{ij}^{rw}$	A binary variable with value 1 if replicated SL $(i, j)$ is used by request $r$ that departs from $i$ at time $p_{ij}^w$ , 0 otherwise, $\forall r \in \mathcal{P}, (i, j) \in \mathcal{F}^r, w \in \mathcal{K}^{\psi^i \psi^j}$

transfer node represented by  $t$  (e.g.  $\psi^{1a} = \psi^{1b} = 1$ ). Consequently,  $\mathcal{T}^t$  can be defined as  $\{i \in \mathcal{T} \mid \psi^i = \psi^t, i \neq t\}$ ,  $\forall t \in \mathcal{T}$  (e.g. in Figure 2(b),  $\mathcal{T}^{2a} = \{2b\}$ , as 2a and 2b represent physical transfer node 2). Note that  $\mathcal{T}^t \subset \mathcal{T}, \forall t \in \mathcal{T}$ .

The proposed model can consider any complex SL network topology. Thus, we also assume that the requests can be transferred from one SL to another. Table 2 shows the parameters used in our model.

The sets can be represented as in Table 3.

Table 4 shows the decision variables used in the proposed formulation.

### 3.2. An arc-based formulation

The PDPTW-SL is formalized as an arc-based MIP, as follows:

$$\min \sum_{(i,j) \in \mathcal{A}} \sum_{v \in \mathcal{V}} \phi_v Y_{ij} x_{ij}^v + \sum_{r \in \mathcal{P}} \sum_{(i,j) \in \mathcal{F}^r} \sum_{w \in \mathcal{K}^{\psi^i \psi^j}} \eta_{ij} d_r q_{ij}^{rw}. \quad (1)$$

Term (1) minimizes the total travel cost of the PD vehicles and the cost of using SLs for the transferred requests.

subject to

The routing and flow constraints

$$\sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{V}} x_{ij}^v = 1 \quad \forall j \in \mathcal{N}_1, \quad (2)$$

$$\sum_{i \in \mathcal{N}_2} x_{o_v, i}^v \leq 1 \quad \forall v \in \mathcal{V}, \quad (3)$$

$$\sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{V}} x_{it}^v \leq 1 \quad \forall t \in \mathcal{T}, \quad (4)$$

$$\sum_{j \in \mathcal{N}_1} x_{ij}^v - \sum_{j \in \mathcal{N}_2} x_{ji}^v = 0 \quad \forall i \in \mathcal{N}, v \in \mathcal{V}, \quad (5)$$

$$\sum_{j \in \mathcal{N}_2} y_{ij}^r - \sum_{j \in \mathcal{N}_2} y_{ji}^r = f_i^r \quad \forall r \in \mathcal{P}, i \in \mathcal{N}_2, \quad (6)$$

$$\sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{V}} x_{it}^v \leq \sum_{r \in \mathcal{P}} \sum_{(i,j) \in \mathcal{F}^t} y_{ij}^r \quad \forall t \in \mathcal{T}, \quad (7)$$

$$\sum_{r \in \mathcal{P}} d_r y_{ij}^r \leq \sum_{v \in \mathcal{V}} Q_v x_{ij}^v \quad \forall (i, j) \in \mathcal{A}^1. \quad (8)$$

In the first block of constraints, (2) assure that all PD nodes are visited exactly once. Constraints (3) make sure that each considered PD vehicle is used at most once, and



constraints (4) avoid visiting each replicated transfer node more than once. Flow balance for the PD vehicles is assured by constraints (5). Constraints (6) enforce flow balance for each request. Constraints (7) assure that a PD vehicle should pick up/drop off a request at a replicated transfer node related to a SL, if there is a flow on this specific SL. Constraints (8) force the capacity of each PD vehicle.

The scheduling constraints

$$y_{ij}^r = 1 \Rightarrow \gamma_j^r \geq \gamma_i^r + Y_{ij} + s_j \quad \forall r \in \mathcal{P}, i, j \in \mathcal{N}_2, \quad (9)$$

$$\sum_{v \in \mathcal{V}} x_{ij}^v = 1 \Rightarrow \beta_j \geq \beta_i + Y_{ij} + s_j \quad \forall i \in \mathcal{N}, j \in \mathcal{N}_2, \quad (10)$$

$$x_{i,o_v}^v = 1 \Rightarrow \alpha_v \geq \beta_i + Y_{i,o_v} + s_{o_v} \quad \forall i \in \mathcal{N}_2, v \in \mathcal{V}, \quad (11)$$

$$\beta_{r+n} \geq \beta_r + Y_{r,r+n} + s_{r+n} \quad \forall r \in \mathcal{P}, \quad (12)$$

$$l_i \leq \beta_i - s_i \leq u_i \quad \forall i \in \mathcal{N}_1, \quad (13)$$

$$l_{g_v} \leq \alpha_v \leq u_{o_v} \quad \forall v \in \mathcal{V}, \quad (14)$$

$$\sum_{w \in \mathcal{K}^{\psi^i \psi^j}} q_{ij}^{rw} = y_{ij}^r \quad \forall r \in \mathcal{P}, (i, j) \in \mathcal{F}^r, \quad (15)$$

$$q_{ij}^{rw} = 1 \quad \text{and} \quad y_{ij}^r = 1 \Rightarrow \gamma_i^r = p_{ij}^w \quad \forall r \in \mathcal{P}, (i, j) \in \mathcal{F}^r, w \in \mathcal{K}^{\psi^i \psi^j}, \quad (16)$$

$$\sum_{r \in \mathcal{P}} \sum_{(a,b) \in \mathcal{F}^{ij}} d_r q_{ab}^{rw} \leq k_{ij} \quad \forall (i, j) \in \mathcal{E}, w \in \mathcal{K}^{ij}. \quad (17)$$

Timing for each request is considered in constraints (9). Similarly for PD vehicles, scheduling is updated in constraints (10) and (11). Constraints (12) assure the precedence constraints, and constraints (13) and (14) enforce the TWs. Constraints (15) and (16) assure the scheduling of the available SLs. Constraints (17) make sure that the package carrying capacity on the SL is not exceeded.

The synchronisation constraints

$$\sum_{j \in \mathcal{N}_1} y_{ij}^r = 1 \Rightarrow \gamma_i^r = \beta_i \quad \forall r \in \mathcal{P}, i \in \mathcal{T}, \quad (18)$$

$$\sum_{j \in \mathcal{N}_2} y_{ij}^r = 1 \Rightarrow \gamma_i^r = \beta_i \quad \forall r \in \mathcal{P}, i \in \mathcal{N}_1, \quad (19)$$

$$\sum_{i \in \mathcal{N}_2} y_{i,r+n}^r = 1 \Rightarrow \gamma_{r+n}^r = \beta_{r+n} \quad \forall r \in \mathcal{P}, \quad (20)$$

$$y_{tj}^r = 1 \Rightarrow \gamma_t^r = \beta_t \quad \forall r \in \mathcal{P}, t \in \mathcal{T}, j \in \mathcal{T}^t. \quad (21)$$

Time synchronization between requests and PD vehicles is considered in constraints (18)–(21). For clarity reasons, we present constraints (9)–(11), (16) and (18)–(21) as implications. However, well-known big-M technique can be used to linearize these constraints.

### Decision variable domains

$$x_{ij}^v \in \{0, 1\} \quad \forall (i, j) \in \mathcal{A}, v \in \mathcal{V}, \quad (22)$$

$$y_{ij}^r \in \{0, 1\} \quad \forall i, j \in \mathcal{N}_2, r \in \mathcal{P}, \quad (23)$$

$$\alpha_v \in R^+ \quad \forall v \in \mathcal{V}, \quad (24)$$

$$\gamma_i^r \in R^+ \quad \forall i \in \mathcal{N}_2, r \in \mathcal{P}, \quad (25)$$

$$\beta_i \in R^+ \quad \forall i \in \mathcal{N}, \quad (26)$$

$$q_{ij}^{rw} \in \{0, 1\} \quad \forall r \in \mathcal{P}, (i, j) \in \mathcal{F}^r, w \in \mathcal{K}^{\psi^i \psi^j}. \quad (27)$$

Since PDPTW-SL is an extension of the PDPTW, it is a non-deterministic polynomial-time (NP)-hard problem. We also note that an optimal solution for the PDPTW represents an upper bound for the PDPTW-SL because the opportunity of using SLs does not restrict the option of direct shipping. The complexity of the PDPTW-SL based on its input is as follows:

- $O(|\mathcal{N}|^2 \cdot |\mathcal{V}|)$ , if  $|\mathcal{V}| \geq n$ ,
- $O(|\mathcal{NO}|^2 \cdot n)$ , if  $|\mathcal{V}| < n$ .

On the other hand, the complexity of the PDPTW based on its input is  $O(|\mathcal{NT}|^2 \cdot |\mathcal{V}|)$ . It is noted that  $|\mathcal{T}| = |\mathcal{S}| \cdot n$  and in most of the cases  $n > |\mathcal{V}|$ . Hence, the complexity of the PDPTW-SL is greater than or equal to the complexity of the PDPTW (i.e.  $O(\text{PDPTW-SL}) \geq O(\text{PDPTW})$ ).

Considering the explosion in complexity along with the increase in instance size, a certain preprocessing needs to be implemented in order to reduce the size of the network. The subsequent section will introduce several elimination rules.

#### 3.2.1. Preprocessing

A preprocessing was implemented in order to reduce the number of decision variables. Hence, some of the infeasible arcs in graph  $\mathcal{G}$  are removed.

- A vehicle cannot leave from and return to a depot other than its own.
- No vehicle can travel from the destination to the origin of the same request.
- No PD vehicle can travel between nodes  $i$  and  $j$ , if  $(i, j) \in \mathcal{F}$ .
- No request  $r$  can travel between origin and destination of a fixed line other than arc  $(i, j) \in \mathcal{F}^r$ .
- No request  $r$  can travel from the destination to the origin of the same request.
- No request is allowed to travel to/from any depot.
- No flow is allowed between nodes  $i$  and  $j$ , if  $l_i + s_i + Y_{ij} > u_j$ .
- No request can travel from its delivery node to any other node.
- No request can travel to its pickup node from any other node.

In addition, some elimination rules related to transfer nodes can be applied.

- An arc  $(t_1, t_2)$  is infeasible if  $l_r + s_r + Y_{r,t_1} + s_{t_1} + Y_{t_1,t_2} + s_{t_2} + Y_{t_2,r+n} > u_{r+n} \quad \forall r \in \mathcal{P}, (t_1, t_2) \in \mathcal{F}^r$ .

The number of departure times on the SLs for each request can be reduced by considering the TWs. For example, no request  $r$  can depart at a time earlier than  $(l_r + s_r + Y_{r,t_1} + s_{t_1})$  on the SL  $(t_1, t_2)$ ,  $\forall r \in \mathcal{P}$ ,  $(t_1, t_2) \in \mathcal{F}^r$ . Similarly, no request  $r$  can depart at a time later than  $(u_{r+n} - s_{r+n} - Y_{t_1,r+n} - s_{t_1} - Y_{t_2,t_1} - s_{t_2})$  on a SL  $(t_2, t_1)$ ,  $\forall r \in \mathcal{P}$ ,  $(t_2, t_1) \in \mathcal{F}^r$ .

### 3.2.2. Model tightening

As the proposed arc-based formulation grows very rapidly in size with the number of requests, number of SLs and departure times on each line, some additional constraints (valid inequalities) may be added to improve the formulation. This section introduces three sets of cuts (i.e. flow tightening, vehicle service tightening and depot strengthening) that are valid for the PDPTW-SL.

We state Proposition 3.1 for flow tightening as follows.

**Proposition 3.1:** *The following inequality is valid for the PDPTW-SL:*

$$\sum_{v \in \mathcal{V}} x_{ij}^v \geq y_{ij}^r \quad \forall (i, j) \in \mathcal{A}^1, r \in \mathcal{P}. \quad (28)$$

**Proof:** Let  $(i, j) \in \mathcal{A}^5$ . In addition, let  $x_{ij}^v$ ,  $\forall v \in \mathcal{V}$ ,  $(i, j) \in \mathcal{A}$  and  $y_{ij}^r$ ,  $\forall r \in \mathcal{P}$ ,  $i, j \in \mathcal{N}_2$  be binary variables. We consider two cases: (1) PD vehicle does not traverse  $(i, j) \in \mathcal{A}^1$  (i.e.  $\sum_{v \in \mathcal{V}} x_{ij}^v = 0$ ) and (2) PD vehicle travels on  $(i, j) \in \mathcal{A}^1$  (i.e.  $\sum_{v \in \mathcal{V}} x_{ij}^v = 1$ ).

In case (1), due to capacity constraints (8), no request can travel on  $(i, j)$  (i.e.  $y_{ij}^r = 0$ ,  $\forall r \in \mathcal{P}$ ). In case (2), any request can travel on  $(i, j)$  as long as constraints (8) are satisfied. Therefore,  $y_{ij}^r \in \{0, 1\}$ ,  $\forall r \in \mathcal{P}$  ensure that  $\sum_{v \in \mathcal{V}} x_{ij}^v \geq y_{ij}^r$ ,  $\forall r \in \mathcal{P}$ .  $\square$

**Example 3.1:** Let there be a flow between  $i$  and  $j$  traversed by a vehicle  $v$  with its capacity equal to two units (i.e.  $Q_v = 2$ ). In addition, two requests,  $r_1$  and  $r_2$ , each having demand one (i.e.  $d_{r_1} = d_{r_2} = 1$ ) travel on the same arc. Assume a linear programming (LP) relaxation solution with following values:  $x_{ij}^v = 0.2$ ,  $y_{ij}^{r_1} = 0.3$  and  $y_{ij}^{r_2} = 0.1$ . Hence, constraints (8) are satisfied ( $0.3 + 0.1 \leq 2 \times 0.2$ ). By using constraints (28),  $x_{ij}^v$  is restricted to be at least 0.3 and  $y_{ij}^{r_1} = 0.3$  and  $y_{ij}^{r_2} = 0.1$ , respectively. The resulting inequality is valid ( $0.3 + 0.1 \leq 2 \times 0.3$ ).

A second proposition for vehicle service tightening is given in Proposition 3.2.

**Proposition 3.2:** *The following inequalities are valid for the PDPTW-SL:*

$$\sum_{i \in \mathcal{N}} x_{i,r}^v - \sum_{i \in \mathcal{N}} x_{r+n,i}^v \leq \sum_{(i,j) \in \mathcal{F}^r} y_{ij}^r \quad \forall r \in \mathcal{P}, v \in \mathcal{V}, \quad (29)$$

$$\sum_{i \in \mathcal{N}} x_{i,r}^v - \sum_{i \in \mathcal{N}} x_{r+n,i}^v \geq - \sum_{(i,j) \in \mathcal{F}^r} y_{ij}^r \quad \forall r \in \mathcal{P}, v \in \mathcal{V}. \quad (30)$$

**Proof:** Let  $r \in \mathcal{P}$  and  $v \in \mathcal{V}$ . In addition, let  $x_{ij}^v$ ,  $\forall v \in \mathcal{V}$ ,  $(i, j) \in \mathcal{A}$  and  $y_{ij}^r$ ,  $\forall r \in \mathcal{P}$ ,  $i, j \in \mathcal{N}_2$  be binary variables. We consider three cases: (1)  $r$  is served by PD vehicle  $v$ ; (2)  $r$  is served by two different PD vehicles,  $v$  and e.g.  $v_1 \in \mathcal{V}$ ; (3)  $r$  is served by other PD vehicle (s)  $v_1$  and/or  $v_2 \in \mathcal{V}$ ;

In case (1), due to constraints (2), pickup ( $\sum_{i \in \mathcal{N}} x_{i,r}^v = 1$ ) and delivery nodes ( $\sum_{i \in \mathcal{N}} x_{r+n,i}^v = 1$ ) of  $r$  are visited by  $v$ , hence the left-hand side (LHS) of the constraints (29) and (30) take a value of zero. Furthermore,  $\sum_{(i,j) \in \mathcal{F}^r} y_{ij}^r \geq 0$  ensures that even request  $r$  is served by  $v$ , it may still use a SL as part of its journey.

In case (2), the pickup or the delivery node of  $r$  is visited by  $v$ . Hence, LHS of constraints (29) and (30) may take two values:  $-1$  and  $1$ . In order to satisfy constraints (7),  $\sum_{(i,j) \in \mathcal{F}^r} y_{ij}^r > 0$ , which implies that  $r$  must use a SL as part of its journey.

In case (3), PD vehicle  $v$  does not visit the pickup, nor the delivery node of  $r$ , thus LHS gets value zero. Hence,  $\sum_{(i,j) \in \mathcal{F}^r} y_{ij}^r \geq 0$  guarantees that  $r$  can use a SL, as it may be served by one or two other PD vehicles. The result follows from these cases.  $\square$

A depot-related valid inequality is given in Proposition 3.3.

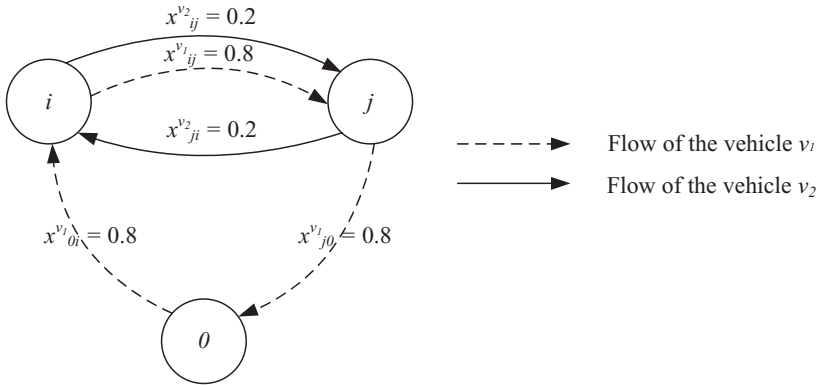
**Proposition 3.3:** *The following inequality is valid for the PDPTW-SL:*

$$M \sum_{j \in \mathcal{N}_2} x_{o_v,j}^v - \sum_{i \in \mathcal{N}_2} \sum_{j \in \mathcal{N}_2} x_{ij}^v \geq 0 \quad \forall v \in \mathcal{V}. \quad (31)$$

**Proof:** Let  $v \in \mathcal{V}$  and  $x_{ij}^v, \forall v \in \mathcal{V}, (i, j) \in \mathcal{A}$  be binary variable. Assume that each vehicle starts and ends its operating day at its depot (i.e.  $o_v$ ). We know that any node  $i \in \mathcal{N}_2$  cannot be visited before leaving the depot. Due to the timing constraints (10) and (11), this assumption holds and the departure of  $v$  from its depot should be smaller than the departure time from any other node. More specifically, if  $\sum_{i \in \mathcal{N}_2} \sum_{j \in \mathcal{N}_2} x_{ij}^v > 0$ , then  $\sum_{j \in \mathcal{N}_2} x_{o_v,j}^v$  gets a value of 1  $\Rightarrow$  inequality (31) holds.  $\square$

**Example 3.2:** Assume a part of an LP solution with two PD vehicles, namely  $v_1, v_2 \in \mathcal{V}$  and  $i, j \in \mathcal{N}_2$  and  $o_{v_1} = o_{v_2} = 0$  (see Figure 3). This LP solution satisfies constraints (2)–(21).

Constraints (31) ensure that  $x_{0i}^{v_1} > 0$  and/or  $x_{0j}^{v_1} > 0$ . Thus, the LP solution given in Figure 3 violates constraints (31). In addition, note that  $M$  can be substituted by  $|\mathcal{N}_2|$ , which implies that a PD vehicle can visit each node in  $\mathcal{N}_2$  at most once.



**Figure 3.** An example of violation for constraint (31).

## 4. Computational results

This section presents the results obtained from solving the proposed arc-based formulation of the PDPTW-SL using CPLEX. The model is implemented in NetBeans IDE 7.1.1 using the corresponding CPLEX 12.1 libraries. Furthermore, all experiments are run on a server with four CPU's (2.4 GHz/6 cores) and 32 GB RAM.

### 4.1. Instance description

We have generated three sets of instances with up to three SLs (triangular topology), namely  $R$ ,  $C$  and  $RC$ , which differ in the geographical distribution of the request nodes. More specifically,  $C$  consists of nodes positioned within at most 30 time units to one of the available three transfer nodes whereas  $RC$  includes nodes within 80 time units, respectively, and  $R$  considers uniform-randomly distributed request nodes. The frequency of the SLs is one departure every 30 time units. Each instance contains up to 12 requests (i.e. 12 pickup and 12 delivery nodes) over  $200 \times 200$  dimensional units on an Euclidean space. In these instances, one distance unit corresponds to one driving time unit, thus the longest trip between two locations would be approximately 4 hours. It would, therefore, represent a service area of a metropolis. Instances are named in  $G\_n\_sl\_v$  format, where  $G$  is the geographic characteristic of the customer nodes,  $n$  is the amount of freight requests that need to be serviced,  $sl$  is the amount of available SLs and  $v$  is the amount of available PD vehicles. In all cases, two depots with up to three heterogeneous PD vehicles each are considered.

We assume a planning horizon of 600 time units (i.e. 10 working hours). The widths of the TWs are generated randomly with an average of 40 time units. Service times are assumed to be up to three time units. The demand of each request is randomly chosen between one and three units. The carrying capacities of PD vehicles is randomly generated between 16 and 20 units. The SL capacity is assumed to be 15 demand units. The instance sets are available on [www.smartlogisticslab.nl](http://www.smartlogisticslab.nl).

Tables 5 and 6 present further details regarding instances. In each table, the column Instance shows the identification of the instance. Table 5 provides the number of depots ( $\mathcal{O}$ ), the number of PD vehicles ( $\mathcal{V}$ ), the number of requests ( $\mathcal{P}$ ), the number of fixed lines ( $\mathcal{E}$ ) and their corresponding scheduled departure times. Table 6 provides the number of constraints, the number of variables in the resulting formulation, time to solve the LP relaxation and finally, objective function value of the LP relaxation. Corresponding values are computed with and without the proposed preprocessing in Section 3.2.1.

We note that the driving cost per time unit of all PD vehicles is assumed to be 0.5 units. It seems reasonable considering operational costs such as fuel consumption, driver wage, insurance and tax. The cost of each package request shipped on every SL is set to 1 unit, which includes handling, storage and transportation costs.

As can be observed from Table 6, the results indicate that there are significant reductions in terms of the number of constraints and decision variables due to preprocessing. The average percentage of reductions for the number of constraints and variables are found to be 18.3% and 20.7%, respectively. Moreover, the improvement of the LP is on average 11.5%. In addition, the preprocessing decreases the number of constraints and variables for the PDPTW by 4.1% and 33.7%, respectively.

**Table 5.** Input sets.

Instance	$ \mathcal{O} $	$ \mathcal{V} $	$ \mathcal{P} $	$ \mathcal{E} $	$ \mathcal{E}  \cdot  \mathcal{K}^{ij} $
C_6_1_4	2	4	6	2	36
C_7_1_4	2	4	7	2	36
C_8_1_4	2	4	8	2	36
C_9_1_4	2	4	9	2	36
C_10_1_4	2	4	10	2	36
C_11_1_4	2	4	11	2	36
C_12_1_4	2	4	12	2	36
RC_6_1_4	2	4	6	2	36
RC_7_1_4	2	4	7	2	36
RC_8_1_4	2	4	8	2	36
RC_9_1_4	2	4	9	2	36
RC_10_1_4	2	4	10	2	36
RC_11_1_4	2	4	11	2	36
RC_12_1_6	2	6	12	2	36
R_6_1_4	2	4	6	2	36
R_7_1_4	2	4	7	2	36
R_8_1_4	2	4	8	2	36
R_9_1_4	2	4	9	2	36
R_10_1_4	2	4	10	2	36
R_11_1_4	2	4	11	2	36
R_12_1_6	2	6	12	2	36

When comparing average numbers of constraints and variables of the PDPTW and PDPTW-SL, it can be observed that the latter model is more complex. On average, there are 85% more constraints and 95% more variables in PDPTW-SL than in the standard PDPTW. In addition, the solution space of the PDPTW-SL is larger than of the PDPTW

**Table 6.** The effect of preprocessing.

Instance	No preprocessing						Preprocessing					
	PDPTW-SL				PDPTW		PDPTW-SL				PDPTW	
	Cons	Vars	LP	CPU	Cons	Vars	Cons	Vars	LP	CPU	Cons	Vars
C_6_1_4	5208	5055	188.72	1.07	910	382	4363	4206	241.18	0.92	892	286
C_7_1_4	7714	7406	205.48	1.41	1199	508	6412	6052	224.29	1.33	1105	376
C_8_1_4	11,069	12,448	220.06	1.43	1528	646	9244	10,199	240.52	1.27	1460	466
C_9_1_4	15,123	16,629	232.57	4.80	1897	788	12,727	13,671	254.59	3.23	1803	552
C_10_1_4	20,059	21,589	237.87	5.74	2306	934	17,008	17,822	262.26	4.04	2156	634
C_11_1_4	26,008	27,485	247.40	23.87	2755	1120	21,590	22,104	267.91	19.12	2671	752
C_12_1_4	33,063	34,321	266.29	45.62	3244	1302	27,702	27,834	291.58	26.28	3175	854
RC_6_1_4	5133	4972	359.73	0.22	910	362	4164	3970	447.18	0.26	891	246
RC_7_1_4	7619	7304	358.80	0.91	1199	488	6169	5770	374.64	0.88	1092	328
RC_8_1_4	10,821	10,249	343.26	1.60	1528	630	8677	7965	366.35	1.55	1422	418
RC_9_1_4	14,753	13,790	355.60	2.42	1897	772	12,129	10,991	388.38	1.82	1752	504
RC_10_1_4	19,660	18,158	374.23	44.73	2306	942	16,177	14,453	399.89	16.61	2224	610
RC_11_1_4	25,429	23,209	409.71	5.79	2755	1112	21,122	18,636	438.84	4.22	2641	708
RC_12_1_6	32,389	29,269	446.61	9.39	4548	1976	26,116	22,731	464.49	7.61	4415	1244
R_6_1_4	5233	5115	312.30	0.30	910	422	3990	3809	367.65	0.29	870	310
R_7_1_4	7737	7457	345.82	0.38	1199	540	6034	5666	410.33	0.41	1123	384
R_8_1_4	10,931	10,373	411.59	1.27	1528	658	8681	8005	445.18	1.17	1416	454
R_9_1_4	14,949	13,988	403.94	1.88	1897	796	11,992	10,864	456.51	1.92	1766	532
R_10_1_4	19,774	18,284	428.26	5.63	2306	966	15,626	13,928	446.32	5.22	2244	634
R_11_1_4	25,535	23,317	449.34	3.36	2755	1124	20,402	17,910	540.75	2.84	2701	716
R_12_1_6	32,290	33,601	518.79	31.07	4548	1910	26,052	26,022	607.09	22.75	4486	1178
Average	338.87						377.90					

because the optimal solution of the PDPTW represents an upper bound for the optimal solution of the PDPTW-SL (due to the flexibility of using SLs).

#### 4.2. CPLEX-standard versus CPLEX with user cuts

In Table 7, we indicate the lower bounds obtained at the root node (after applying preprocessing). Column LP shows the bounds obtained without applying any kind of cut generation. CPX shows the bounds obtained by activating automatic cut generation of CPLEX (without user-defined cuts). Finally, Full column shows the bounds obtained by applying both CPLEX and user-defined cuts. CPU column shows the time needed to get the improved lower bound using CPLEX and user-defined cuts. The last column indicates the improvement in the lower bound when using user cuts with CPLEX cuts at a time, compared to CPLEX cuts alone.

Tables 8 and 9 report the results obtained by using CPLEX with and without proposed cuts. In both cases, a time limit of 36,000 seconds (10 hours) was imposed to the solver. We indicate the objective function value, obtained GAP from the best bound, the best bound obtained, the CPU time (in seconds) and the number of explored nodes during the search process. Note that ‘–’ means no integer solution was found within the imposed time limit.

Computational results obtained in this section show that instances with up to 11 requests can be solved to optimality in a reasonable time by using the proposed model with the additional constraints introduced in Section 3.2.2. One can observe that taking advantage of the proposed cuts significantly improves the performance of CPLEX.

A total number of 10 instances were solved to optimality by both variants of the model within the time limit. On average, the CPU time was 4437 seconds for CPLEX alone

**Table 7.** Initial lower bounds.

Instance	LP	CPX	Full	CPU	Improvement (%)
C_6_1_4	241.18	259.49	260.21	9.69	0.28
C_7_1_4	224.29	264.48	265.70	21.78	0.46
C_8_1_4	240.52	277.00	279.78	48.49	1.00
C_9_1_4	254.59	283.95	301.68	63.22	6.24
C_10_1_4	262.26	289.85	310.26	230.84	7.04
C_11_1_4	267.91	320.00	320.63	130.37	0.20
C_12_1_4	291.58	319.66	345.59	582.94	8.11
RC_6_1_4	447.18	487.68	487.76	3.83	0.02
RC_7_1_4	374.64	476.33	483.61	11.35	1.53
RC_8_1_4	366.35	475.43	481.60	23.12	1.30
RC_9_1_4	388.38	496.27	499.96	16.98	0.74
RC_10_1_4	399.89	489.07	503.43	52.25	2.94
RC_11_1_4	438.84	531.21	539.44	63.31	1.55
RC_12_1_6	464.49	564.20	571.24	83.67	1.25
R_6_1_4	367.65	412.01	416.16	0.43	1.01
R_7_1_4	410.33	426.34	439.02	0.79	2.97
R_8_1_4	445.18	481.26	504.35	7.56	4.80
R_9_1_4	456.51	492.14	514.09	33.45	4.46
R_10_1_4	446.32	497.07	515.62	41.98	3.73
R_11_1_4	540.75	602.04	622.92	32.62	3.47
R_12_1_6	607.09	671.09	681.27	97.57	1.52
	377.90	434.12	444.96		2.60

**Table 8.** Results using CPLEX without user cuts.

Instance	Solution value	GAP (%)	Bound	Nodes	CPU (sec)
C_6_1_4	369.36	0.00	369.36	37,512	4341
C_7_1_4	390.31	0.00	390.30	33,633	5142
C_8_1_4	446.35	17.93	366.31	79,476	36,000
C_9_1_4	—	—	317.36	131,527	36,000
C_10_1_4	—	—	331.17	19,517	36,000
C_11_1_4	—	—	329.97	26,774	36,000
C_12_1_4	—	—	354.12	7399	36,000
RC_6_1_4	572.76	0.00	572.76	12,708	404
RC_7_1_4	575.95	0.00	575.95	70,057	2210
RC_8_1_4	585.32	0.00	585.32	310,184	13,392
RC_9_1_4	—	—	504.02	284,774	36,000
RC_10_1_4	—	—	512.08	104,181	36,000
RC_11_1_4	—	—	547.57	132,419	36,000
RC_12_1_6	—	—	578.09	107,558	36,000
R_6_1_4	416.16	0.00	416.16	981	1
R_7_1_4	473.06	0.00	473.06	295	1
R_8_1_4	558.17	0.00	558.17	980	35
R_9_1_4	632.42	0.00	632.42	14,941	2116
R_10_1_4	636.05	0.00	636.05	200,928	16,729
R_11_1_4	748.29	11.61	661.39	16,9960	36,000
R_12_1_6	—	—	695.53	67,232	36,000

compared with 1105 seconds for CPLEX with additional cuts. The average number of explored nodes drops from 68,222 to 6879. Proposed cuts helped in proving optimality within imposed time limit for the four instances that CPLEX without user-cuts failed to solve. For the data-sets that were not solved to optimality by CPLEX alone, the user cuts helped obtaining tighter bounds or proving optimality.

**Table 9.** Results using CPLEX with user cuts.

Instance	Solution value	GAP (%)	Bound	Nodes	CPU (sec)
C_6_1_4	369.36	0.00	369.36	20,790	3334
C_7_1_4	390.31	0.00	390.30	20,789	3762
C_8_1_4	446.35	13.76	384.93	47,787	36,000
C_9_1_4	493.66	33.67	327.43	24,290	36,000
C_10_1_4	—	—	335.89	9081	36,000
C_11_1_4	—	—	347.68	4278	36,000
C_12_1_4	—	—	366.81	2100	36,000
RC_6_1_4	572.76	0.00	572.76	2036	61
RC_7_1_4	575.95	0.00	575.95	2208	161
RC_8_1_4	585.32	0.00	585.32	3142	510
RC_9_1_4	593.70	0.00	593.70	70,292	32,204
RC_10_1_4	599.95	0.00	599.95	37,019	24,925
RC_11_1_4	624.48	0.00	624.48	12,879	18,128
RC_12_1_6	—	—	608.89	8281	36,000
R_6_1_4	416.16	0.00	416.16	233	0
R_7_1_4	473.06	0.00	473.06	62	1
R_8_1_4	558.17	0.00	558.17	458	10
R_9_1_4	632.42	0.00	632.42	12,343	1915
R_10_1_4	636.05	0.00	636.05	6728	1295
R_11_1_4	748.29	0.00	748.29	46,044	17,493
R_12_1_6	—	—	771.01	17,878	36,000



### 4.3. Scenarios

In order to quantify the benefits of the proposed transportation system, we define and compare two scenarios. The first one considers proposed integrated transportation system (PDPTW-SL). The second scenario considers standard PD system (PDPTW). CPLEX time limit is set to 24 hours and the best solution values are shown in Table 10.

We compare the scenarios by investigating: (1) total cost of the service provider and (2) total driving time by PD vehicles.

The PDPTW-SL proved, on average, 9% cost savings compared to the standard PDPTW. Note that for some  $R$  instances no savings were achieved as no packages were transferred to the SLs. In this case, the optimal solution for the PDPTW is also found to be optimal for the PDPTW-SL. Overall, along with the increase in number of requests, the cost savings tend to remain significant (i.e.  $C$ ,  $RC$  instances).

It is important to note that the effectiveness of the proposed system can be highly dependent on both, the spatial pattern of the requests and the configuration of SLs. Although public transportation is integrated with vehicle routing, in some cases no benefits can be achieved from an integrated system, as observed in some of  $R$  instances. Therefore, designing such a system involves tactical decisions related to the pattern of the SLs (positioning of the transfer nodes relative to the demand cluster centres), the storage areas at the transfer nodes, and the re-design of the SL vehicles (e.g. extra freight capacity), that are not taken into consideration in this paper (the focus is on operational costs of the system).

**Table 10.** A comparison between PDPTW and PDPTW-SL.

Instance	Total costs			Total driving time		
	PDPTW-SL	PDPTW	Savings (%)	PDPTW-SL	PDPTW	Savings (%)
C_6_1_4	369.36	461.70	20.0	728.7	923.40	21.1
C_7_1_4	390.31	464.49	16.0	768.59	928.98	17.3
C_8_1_4	446.35	483.50	—	882.67	967.00	8.7
C_9_1_4	<b>471.16</b>	498.19	—	<b>932.30</b>	996.37	—
C_10_1_4	<b>600.53</b>	515.25	—	<b>1189.06</b>	1030.49	—
C_11_1_4	—	524.69	—	—	1049.38	—
C_12_1_4	—	598.55	—	—	1197.10	—
RC_6_1_4	572.76	658.99	13.1	1143.50	1317.98	13.2
RC_7_1_4	575.95	662.18	13.0	1149.88	1324.36	13.2
RC_8_1_4	585.32	663.16	11.7	1168.63	1326.32	11.9
RC_9_1_4	593.70	703.30	15.6	1181.38	1407.00	16.0
RC_10_1_4	599.95	709.55	15.4	1193.88	1419.10	15.9
RC_11_1_4	624.48	742.52	15.9	1242.94	1485.04	16.3
RC_12_1_6	—	794.42	—	—	1588.84	—
R_6_1_4	416.16	416.16	0.0	832.32	832.32	0.0
R_7_1_4	473.06	473.06	0.0	946.12	946.12	0.0
R_8_1_4	558.17	558.17	0.0	1116.34	1116.34	0.0
R_9_1_4	632.42	653.12	3.2	1258.82	1306.24	3.6
R_10_1_4	636.05	658.60	3.4	1266.08	1317.20	3.9
R_11_1_4	748.29	752.38	0.5	1490.55	1504.76	0.9
R_12_1_6	—	936.23	—	—	1872.44	—

Note: The values in bold indicate that CPLEX could not solve the instances to optimality within imposed time-limit.

**Table 11.** The effect of the number of available SLs on the objective value function.

Instance	One SL			Two SLs			Three SLs		
	UB	GAP (%)	LB	UB	GAP (%)	LB	UB	GAP (%)	LB
C_6	369.36	0.00	369.36	369.36	0.00	369.36	310.84	21.29	244.66
C_7	390.31	0.00	390.31	381.20	13.10	331.26	384.13	46.60	205.13
C_8	446.35	13.76	384.93	—	—	310.83	559.19	59.52	226.36
C_9	471.16	13.25	408.72	—	—	307.60	714.65	66.62	238.56
C_10	600.53	44.07	335.89	—	—	303.28	—	—	244.95
C_11	—	—	349.96	—	—	315.35	—	—	268.55
C_12	—	—	357.40	—	—	340.11	—	—	290.84
RC_6	572.76	0.00	572.76	572.76	0.00	572.76	520.10	0.00	520.10
RC_7	575.95	0.00	575.95	575.95	0.00	575.95	539.08	0.00	539.08
RC_8	585.32	0.00	585.32	585.32	0.00	585.32	548.45	16.89	455.82
RC_9	593.70	0.00	593.70	593.70	0.00	593.70	—	—	435.53
RC_10	599.95	0.00	599.95	—	—	512.73	—	—	442.22
RC_11	624.48	0.00	624.48	—	—	502.54	—	—	448.26
RC_12	—	—	608.89	—	—	504.78	—	—	475.81
R_6	416.16	0.00	416.16	416.16	0.00	416.16	416.16	0.00	416.16
R_7	473.06	0.00	473.06	473.06	0.00	473.06	470.15	0.00	470.14
R_8	558.17	0.00	558.17	535.27	0.00	535.27	490.77	0.00	490.77
R_9	632.42	0.00	632.42	579.38	0.00	579.38	647.27	20.49	514.64
R_10	636.05	0.00	636.05	584.89	0.00	584.89	584.89	4.42	559.06
R_11	748.29	0.00	748.29	687.36	4.10	659.36	—	—	571.73
R_12	—	—	795.18	—	—	683.56	—	—	627.47

#### 4.4. The effect of the number of SLs on the operating costs

In this section, we evaluate the effect of the number of available SLs on the solution cost. A time limit of 24 hours has been imposed to CPLEX for the following experiments and the best solutions found are given in Table 11.

The results indicate an intuitive insight: the more SLs are considered, the more shipments on SLs are made. For the cases with optimal solutions, the average cost savings for the 2-SLs and 3-SLs are 2.4% and 6.2%, compared to the 1-SL case. Overall, the increase in savings can be up to 9% (i.e. RC\_6\_1\_4) when comparing 1-SL case with 3-SLs case. We note that more available SLs lead to higher complexity. In other words, significantly fewer instances with three SLs could be solved, compared to, e.g. 1- or 2-SLs cases.

#### 4.5. The effect of the time windows on the operating costs

In this section, we investigate the effect of the TW tightness on the operational costs. Each pickup or delivery node is assigned a TW  $[l_i^1, u_i^1]$ , such that  $(u_i^1 - l_i^1) = (u_i - l_i)\rho$  and  $|u_i^1 - u_i| = |l_i^1 - l_i|$ , where  $\rho \in \{0.5, 1.5, 2\}$ . A time limit of 24 hours has been imposed to CPLEX for the following experiments. The results are shown in Tables 12 and 13.

According to the obtained results, we can conclude that the more relaxed TWs are, the lower operating costs can be achieved. For instance, in the investigated cases, on average 3.7% (i.e. for 50% wider TWs) and 6.2% (i.e. 100% wider TWs) operating cost savings can be obtained if the TWs are wider compared to the original case (i.e.  $\rho = 1$ ). The case with 50% tighter TWs led to an average of 7.7% deterioration in the objective function value.

However, by comparing the operating costs of the PDPTW-SL optimal solutions to the corresponding PDPTW solutions in all above cases (i.e.  $\rho \in \{0.5, 1.5, 2\}$ ), savings due to the use of available SLs still remain significant (i.e. approximately 7%).

**Table 12.** The effect of the time windows on the operating costs.

Instance	$\rho = 0.5$				$\rho = 1$			
	UB	GAP (%)	LB	PDPTW	UB	GAP (%)	LB	PDPTW
C_6_1_4	458.05	0.00	458.05	470.56	369.36	0.00	369.36	461.70
C_7_1_4	463.72	0.00	463.72	478.65	390.30	0.00	390.30	464.49
C_8_1_4	480.01	0.00	480.01	541.38	446.35	0.00	446.35	483.50
C_9_1_4	509.34	0.00	509.34	556.16	471.16	13.25	408.72	498.19
C_10_1_4	—	—	413.86	667.58	600.53	44.07	335.89	515.25
C_11_1_4	753.73	45.03	414.32	670.30	—	—	349.96	524.69
C_12_1_4	—	—	403.86	737.40	—	—	357.40	598.55
RC_6_1_4	572.76	0.00	572.76	658.99	572.76	0.00	572.76	658.99
RC_7_1_4	575.95	0.00	575.95	662.18	575.95	0.00	575.95	662.18
RC_8_1_4	585.72	0.00	585.72	663.16	585.32	0.00	585.32	663.16
RC_9_1_4	594.09	0.00	594.09	724.88	593.70	0.00	593.70	703.30
RC_10_1_4	600.35	0.00	600.35	731.13	599.95	0.00	599.95	709.55
RC_11_1_4	624.88	0.00	624.88	773.34	624.48	0.00	624.48	742.52
RC_12_1_6	662.65	0.00	662.65	825.20	—	—	608.89	794.42
R_6_1_4	493.05	0.00	493.05	493.05	416.16	0.00	416.16	416.16
R_7_1_4	588.94	0.00	588.94	588.96	473.06	0.00	473.06	473.06
R_8_1_4	639.73	0.00	639.73	639.73	558.17	0.00	558.17	558.17
R_9_1_4	711.72	0.00	711.72	711.72	632.42	0.00	632.42	653.12
R_10_1_4	715.72	0.00	715.72	715.84	636.05	0.00	636.05	658.60
R_11_1_4	802.84	0.00	802.84	846.31	748.29	0.00	748.29	752.38
R_12_1_6	1109.46	0.00	1109.46	1176.35	—	—	795.18	936.23

In addition, all instances with six to seven requests and one SL could be optimally solved without considering the TWs of the requests. The obtained results (optimal solution costs) are shown in Table 14. On average, 5.1% savings in terms of operating costs can be achieved by using available SL compared to the PDP environment, with a maximum of 10% (i.e. RC\_6\_1\_4). Therefore, based on the obtained results, we can conclude that TWs play an important role in the potential savings obtained by the PDPTW-SL

**Table 13.** The effect of the time windows on the operating costs.

Instance	$\rho = 1.5$				$\rho = 2$			
	UB	GAP (%)	LB	PDPTW	UB	GAP (%)	LB	PDPTW
C_6_1_4	362.28	0.00	362.28	366.57	331.69	0.00	331.69	347.68
C_7_1_4	365.07	0.00	365.07	374.66	347.58	0.00	347.58	350.47
C_8_1_4	384.08	0.00	384.08	388.25	365.72	17.11	303.15	369.47
C_9_1_4	422.37	19.13	341.57	481.41	415.26	28.32	297.68	398.49
C_10_1_4	—	—	314.27	498.47	—	—	300.92	432.53
C_11_1_4	—	—	320.39	505.62	—	—	303.15	432.75
C_12_1_4	—	—	340.34	551.50	—	—	323.13	484.90
RC_6_1_4	572.76	0.00	572.76	658.99	547.75	0.00	547.75	630.45
RC_7_1_4	575.95	0.00	575.95	662.18	558.97	0.00	558.97	633.65
RC_8_1_4	583.60	0.00	583.60	663.15	568.35	0.00	568.35	634.62
RC_9_1_4	591.98	0.00	591.98	698.31	591.98	20.29	471.87	671.94
RC_10_1_4	714.18	27.47	518.03	704.57	—	—	458.38	679.95
RC_11_1_4	—	—	569.06	742.52	—	—	481.02	713.98
RC_12_1_6	—	—	571.59	794.42	—	—	511.02	769.79
R_6_1_4	416.16	0.00	416.16	416.16	397.16	0.00	397.16	397.16
R_7_1_4	468.02	0.00	468.02	468.02	411.77	0.00	411.77	411.77
R_8_1_4	552.51	0.00	552.51	558.17	535.32	0.00	535.32	558.17
R_9_1_4	571.74	0.00	571.74	571.74	571.74	13.99	491.76	571.74
R_10_1_4	575.85	0.00	575.85	575.85	575.85	7.93	530.20	575.85
R_11_1_4	711.68	7.67	657.11	711.68	—	—	597.38	707.53
R_12_1_6	—	—	684.08	837.14	—	—	619.05	837.14

**Table 14.** The effect of the time windows on the operating costs.

Instance	PDPTW-SL	PDPTW
C_6_1_4	214.46	238.13
C_7_1_4	231.69	246.81
RC_6_1_4	399.16	445.84
RC_7_1_4	427.15	448.00
R_6_1_4	326.64	326.64
R_7_1_4	377.66	377.66
Average	329.53	357.49

solutions compared to the classical PDPTW. However, no relationship could be observed between the TWs wideness and operating cost savings when comparing the PDPTW-SL and the PDPTW. Note that disregarding TWs for the PDPTW-SL can lead to an average drop of 29% in operating costs, compared to the original case (i.e.  $\rho = 1$ ).

#### 4.6. The effect of the SLs frequency on the operating costs

In this section, we investigate the effect of the departure frequency of the available SLs on the operating costs. We consider three cases, namely the original frequency (once every 30 time units), a departure every 7 time units and, finally, every 180 time units. The obtained results are shown in Table 15.

According to the obtained results, we can conclude another intuitive insight: less frequent SL services lead to higher operating costs, with the PDPTW optimal solution cost as an upper bound.

**Table 15.** The effect of the SLs frequency on the operating costs.

Instance	7 time units	30 time units	180 time units	PDPTW
C_6_1_4	369.36	369.36	441.42	461.70
C_7_1_4	390.31	390.31	452.82	464.49
C_8_1_4	<b>441.66</b>	446.34	483.49	483.50
C_9_1_4	<b>504.87</b>	<b>471.15</b>	498.19	498.19
C_10_1_4	<b>579.81</b>	<b>600.53</b>	510.01	515.25
C_11_1_4	—	—	<b>524.69</b>	524.69
C_12_1_4	—	—	—	598.55
RC_6_1_4	572.76	572.76	658.98	658.99
RC_7_1_4	575.95	575.95	662.18	662.18
RC_8_1_4	585.32	585.32	663.15	663.16
RC_9_1_4	593.70	593.70	703.30	703.30
RC_10_1_4	599.95	599.95	709.55	709.55
RC_11_1_4	624.48	624.48	742.52	742.52
RC_12_1_6	—	—	794.42	794.42
R_6_1_4	416.16	416.16	416.16	416.16
R_7_1_4	473.06	473.06	473.06	473.06
R_8_1_4	558.17	558.17	558.17	558.17
R_9_1_4	632.42	632.42	632.42	653.12
R_10_1_4	636.05	636.05	636.05	658.60
R_11_1_4	748.29	748.29	748.29	752.38
R_12_1_6	—	—	913.68	936.23
Average	555.42	555.42	607.00	612.67

Note: The numbers in bold indicate suboptimal values and '—' is used to show that no integer solution was found within a time limit of 24 hours.

## 5. Conclusions and the future research directions

This paper presented an extension to the PDPTW in the context of package transportation with the opportunity to use public scheduled transportation. A mixed-integer formulation for the PDPTW-SL was proposed. However, due to high computational complexity, the PDPTW-SL can be optimally solved only for small instances. By using the proposed arc-based model, two scenarios were analysed in terms of operating costs.

The proposed transportation system leads to significant operating cost reductions compared to the standard PDPTW environment. Overall, our paper has demonstrated that the operational costs can be reduced by up to 20% by making use of public transportation for carrying small packages.

The proposed model is yet essential for further research. In terms of solution methodologies, a potential research direction could be developing a specialized B&C algorithm for the current arc-based formulation by using the cuts existing in the literature along with the ones proposed in this paper. In addition, meta-heuristic algorithms that generate good-quality solutions for reasonable-sized instances could be investigated. Moreover, as the considered transportation environment is subject to changes (e.g. traffic jams, new requests, etc.), algorithms for solving the dynamic version of the proposed problem could be studied as well. The model can be extended in terms of realistic aspects, such as time-dependent travel times, uncertainty in the input data and driver-related constraints.

## Acknowledgements

The authors would like to thank the two anonymous reviewers for their useful comments and for raising interesting points for discussion.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

The authors gratefully acknowledge funds provided by Dinalog, the Dutch Institute for Advanced Logistics, under the grant titled ‘Cargo Hitching’, #2011 4 086R.

## References

- Aldaihani M, Dessouky M. 2003. Hybrid scheduling methods for paratransit operations. *Comput Ind Eng*. 45: 75–96.
- Berbeglia G, Cordeau J-F, Gribkovskaia I, Laporte G. 2007. Static pickup and delivery problems: a classification scheme and survey. *TOP: an Official Journal of the Spanish Society of Statistics and Operations Research*. 15: 1–31.
- Cordeau J-F, Laporte G. 2007. The dial-a-ride problem: models and algorithms. *Ann Oper Res*. 153: 29–46.
- Cortes E-C, Matamala M, Contardo C. 2010. The pickup and delivery problem with transfers: formulation and a branch-and-cut solution method. *Eur J Oper Res*. 200: 711–724.
- DHL-Packstation. 2013. DHL official webpage [accessed 2015 Nov 19]. Available from: [www.dhl.de/en/paket/pakete-empfangen/packstation.html](http://www.dhl.de/en/paket/pakete-empfangen/packstation.html).

- Häll C, Andersson H, Lundgren J, Varbrand P. 2009. The integrated dial-a-ride problem. *Public Transp.* 1: 39–54.
- Hurtigruten. 2013. Hurtigruten official webpage [accessed 2015 Dec 11]. Available from: [www.hurtigruten-web.com/index\\_en.html](http://www.hurtigruten-web.com/index_en.html).
- Levin Y, Nediak M, Topaloglu H. 2012. Cargo capacity management with allotments and spot market demand. *Oper Res.* 60: 351–365.
- Liaw C-F, White C, Bander J. 1996. A decision support system for the bimodal dial-a-ride problem. *IEEE Trans Syst Man Cybern A.* 26: 552–565.
- Lindholm M, Behrends S. 2012. Challenges in urban freight transport planning – a review in the Baltic Sea Region. *J Transp Geogr.* 22: 129–136.
- Masson R, Lehuède F, Peton O. 2012. An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transp Sci.* 47: 1–12.
- Masson R, Lehuède F, Peton O. 2014. The dial-a-ride problem with transfers. *Comput Oper Res.* 41: 12–23.
- Nash C, editor. 1982. *Economics of public transport*. London: Longman.
- Parragh S, Cordeau J-F, Doerner K, Hartl R. 2012. Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *Oper Res Spectr.* 34: 593–633.
- Parragh SN. 2010. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Trans Res C.* 19: 912–930.
- Petersen HL, Røpke S. 2011. The pickup and delivery problem with cross-docking opportunity. In: Böse JW, Hu H, Jahn C, Shi X, Stahlbock R, Voß S, editors. *Computational logistics*. Vol. 6971. Berlin: Springer; p. 101–113.
- Shang J, Cuff C. 1996. Multicriteria pickup and delivery problem with transfer opportunity. *Comput Ind Eng.* 30: 631–645.
- Trentini A, Malhène N. 2010. Toward a shared urban transport system ensuring passengers & goods cohabitation. *TeMA: J Land Use, Mob Environ.* 4: 37–44.
- Trentini A, Masson R, Lehuède F, Malhène N, Peton O, Tlahig H. 2012. A shared “passenger & goods” city logistics system. In: 4th International Conference on Information Systems, Logistics and Supply Chain, 2012 August; Quebec, Canada.