Transportation Research Part B 91 (2016) 34–51

Contents lists available at ScienceDirect

Transportation Research Part B

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

A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands

Veaceslav Ghilas*, Emrah Demir, Tom Van Woensel

School of Industrial Engineering, Operations, Planning, Accounting and Control (OPAC), Eindhoven University of Technology, Eindhoven, 5600 MB, The Netherlands

ARTICLE INFO

Article history:
Received 9 September 2015
Revised 22 April 2016
Accepted 23 April 2016
Available online 13 May 2016

Keywords:
Freight transportation
Pickup and delivery problem
Scheduled lines
Stochastic demands
Heuristic algorithm

ABSTRACT

The Pickup and Delivery Problem with Time Windows, Scheduled Lines and Stochastic Demands (PDPTW-SLSD) concerns scheduling a set of vehicles to serve a set of requests, whose expected demands are known in distribution when planning, but are only revealed with certainty upon the vehicles' arrival. In addition, a part of the transportation plan can be carried out on limited-capacity scheduled public transportation line services. This paper proposes a scenario-based sample average approximation approach for the PDPTW-SLSD. An adaptive large neighborhood search heuristic embedded into sample average approximation method is used to generate good-quality solutions. Computational results on instances with up to 40 requests (i.e., 80 locations) reveal that the integrated transportation networks can lead to operational cost savings of up to 16% compared with classical pickup and delivery systems.

1. Introduction

Integrated public and freight transportation flows have been successfully achieved in long-haul transportation. Apart from the passengers, cruise ships carry freight between ports (Hurtigruten, 2013). Passenger aircraft are used to transport freight without affecting the service level for passengers (Levin et al., 2012). On the other hand, freight and passenger transportation are rarely integrated in short-haul transportation operations. There are only a few initiatives to investigate the potential benefits of integrated system. MULI was a demonstration project in the west of Germany between 1996 and 1999 (Trentini and Malhene, 2010). Special-design buses were used for both passengers and small-sized packages to reduce emissions. City Cargo Amsterdam was set up as a pilot experiment in 2007 (CargoTram, 2007). Two cargo trams were redesigned to transport packages in the city center of Amsterdam. In 2009, the project was abandoned due to lack of public funds. Later, Masson et al. (2015) investigated a freight distribution system in the French city La Rochelle. The results demonstrated that integrated short-haul transportation systems could lead to improved vehicle utilization and reduced operational costs. In our view, aforementioned projects did not succeed due to a number of practical challenges, such as decision support, information sharing between involved parties and insufficient funding for (re-)designing existing transportation systems.

As a result of economic development and increased populations, urban areas are densely covered by public transportation lines (i.e., bus, metro, tram), which typically operate according to predefined routes and schedules. In this paper, such public

* Corresponding author. Tel.: +31 40 247 4984.
E-mail addresses: V.Ghilas@tue.nl (V. Ghilas), E.Demir@tue.nl (E. Demir), T.v.Woensel@tue.nl (T.V. Woensel).

http://dx.doi.org/10.1016/j.trb.2016.04.015
0191-2615/

transportation services are referred as *scheduled line services* (SLs). A possible integration of freight transportation with SLs might provide efficiency gains for all transport stakeholders. The integrated system can potentially reduce operational costs of logistic service providers (LSPs) and can lead to extra revenue for public transport companies. As a side effect, fewer emissions can be achieved with shorter vehicle routes (Demir et al., 2014; 2015b, see, e.g.,). Integrated transport systems may also be beneficial to low-density areas, where vehicles usually make fewer stops over longer distances (Santos et al., 2010).

This paper investigates the *Pickup and Delivery Problem with Time Windows, Scheduled Lines and Stochastic Demands* (PDPTW-SLSD), in which a set of geographically-spread freight requests need to be transported to their destinations using a fleet of heterogeneous pickup and delivery (PD) vehicles (i.e., a priori optimization). Moreover, capacitated SL services can be used as a part of requests' journey without affecting passenger service level. This characteristic of the PDPTW-SLSD implies that a request can be served in two ways, direct delivery or transferred through SLs. Furthermore, the exact quantities demanded by each customer are only learned upon vehicles' arrival at the corresponding pickup locations. Depending on the demand realizations, there are two possible violation outcomes: *(i)* PD vehicle may arrive at a pickup location without enough carrying capacity, and *(ii)* SL service capacity might not be sufficient for the realized demand. In such cases, called *route failures*, corrective (or recourse) actions need to be applied in order to recover feasibility. These actions obviously lead to extra costs which can be charged by LSPs or freight carriers.

In order to consider demand uncertainty, a *Sample Average Approximation* (SAA) method along with an adaptive large neighborhood search (ALNS) algorithm is proposed. The SAA is a scenario-based framework to deal with stochastic discrete optimization problems (Kleywegt et al., 2001). The basic steps of the SAA are as follows: *(i)* solving the SAA problem for a given restricted set of scenarios (i.e., a subset of a larger set of scenarios), *(ii)* evaluating the found solution on a larger set of scenarios and approximating the expected value function by the sample average function, and *(iii)* iterating (*step (i)–(ii)*) until the defined stopping criterion is reached. We note that exact as well as heuristic algorithms (i.e., ALNS) can be used to solve the SAA problem in *step (i)*.

The scientific contribution of this study is two-fold: (*i*) we propose a solution approach for the integrated short-haul transportation problem in a stochastic environment, and (*ii*) we quantify the benefits of the integrated system in stochastic setting. The remainder of this paper is organized as follows. Section 2 provides a brief literature review on related topics. The problem description and the mathematical formulation are discussed in Section 3. The solution approach is given in Section 4. Section 5 presents the results of extensive computational experiments. Conclusions are stated in Section 6.

2. Literature review

In this section, we present a brief review on the existing literature related to the PDPs with transfers, as these are the most closely related to the PDPTW-SLSD. We then present recent works on stochastic PDPs.

2.1. PDPs with transfers

To the best of our knowledge, there are only a few studies on PDPs with transfers to public transportation services. One of the first attempts to combine the deterministic dial-a-ride problem (DARP) with SLs was done by Liaw et al. (1996). The authors proposed heuristic algorithms to tackle static and dynamic aspects of the problem and solved instances with up to 120 requests. In another study, Aldaihani and Dessouky (2003) investigated an integrated DARP with public transportation. A two-stage heuristic algorithm was proposed to solve instances with up to 155 requests. They concluded that shifting some requests to available SLs reduces the overall trip times of the requests and total traveled distance by all PD vehicles. Häll et al. (2009) formalized the integrated DARP as an arc-based mixed-integer program (MIP) and used an all-purpose solver to tackle instances with up to four requests. The authors disregarded the schedules of the public transportation services.

In another study, Ghilas et al. (2016c) provided an arc-based MIP formulation for the deterministic PDPTW-SL. The authors optimally solved instances with up to 11 requests and showed that significant operating-cost savings of up to 20% can be achieved due to the use of available SLs. As a follow-up study, Ghilas et al. (2016b) proposed an ALNS to solve large-size PDPTW-SL instances. The authors concluded that the benefits of the integrated system remain significant for large instances as well. Later, Ghilas et al. (2016a) developed an exact branch-and-price algorithm to solve small- to medium-size instances of the PDPTW-SL. The authors reformulated the problem in terms of paths (routes), and the latter were generated by solving a variant of the elementary shortest path with resource constraints using a tailored labeling algorithm. Instances with up to 50 requests were optimally solved.

The interested readers are referred to Drexl (2012) for an extensive literature review on Vehicle Routing Problems (VRPs) with synchronization constraints. In addition, a related survey on planning semi-flexible transit systems can be found in Errico et al. (2013).

2.2. Stochastic PDPs

Over the years, there has been an increasing interest for the VRP with stochastic demands (VRPSD) (see, e.g., Laporte et al., 2002; Verweij et al., 2003; Secomandi and Margot, 2009). The most advanced solution approaches can be categorized into two groups according to the use of stochastic techniques. The most commonly used approach is *stochastic programming* *with recourse*, which is a well-known framework for modeling optimization problems that involve uncertainty. Since some data is unknown at the moment of planning, one decision is made and expected (or recourse) costs of the consequences of the plan are minimized. The second most-used approach and the one related to this paper is the *multi-scenario approach* (or Monte Carlo simulation). This method is a common way to estimate expected costs when a closed-form expression is not available. Therefore, an approximation is made by evaluating a solution on a set of generated scenarios. In order to implement a multi-scenario stochastic optimization approach, metaheuristic algorithms can be efficiently engaged with sampling approaches. The interested readers are referred to Bianchi et al. (2009) and Gutjahr (2011) for metaheuristic algorithms proposed in the domain of stochastic combinatorial optimization.

To the best of our knowledge, limited literature exists on PD-related problems (e.g., one-to-one PDP, DARP) with stochastic demands. One of the first works was studied by Powell et al. (1988), who considered the dynamic PDP with stochastic demands in the context of long haul freight transportation. The computational results indicated that substantial profits can be achieved by considering uncertainty in the planning process while increasing the service level compared with the previous deterministic planning approach. Other examples are the single vehicle PDP (or 1-TSP) with stochastic demands by Louveaux and Gonzalez (2009), the DARP with stochastic customers' delays by Heilporn et al. (2011), and the stochastic DARP with expected return transports by Schilde et al. (2011).

3. Problem description and mathematical formulation

The PDPTW-SLSD is an extension of the classical PDPTW (see, e.g., Dumas et al., 1991; Nanry and Barnes, 2000; Lu and Dessouky, 2006). The PDPTW-SLSD consists of routing and scheduling a set of vehicles to transport a set of geographically-spread requests from corresponding origins to their destinations. In addition, a set of limited-capacity scheduled line vehicles operates according to pre-defined routes and schedules. Each request may use any of the available SL services as a part of its journey. In other words, a request may be collected by one PD vehicle, transferred to a scheduled line and afterwards delivered to its final destination after being re-collected by another PD vehicle. Furthermore, each request has to be served within its corresponding time windows.

It is assumed that demand of each request is unknown by the time of planning. The exact quantities demanded by each request are only learned upon the vehicle's arrival at the corresponding pickup locations. However, the demand of each request is assumed to follow a known probability distribution.

Due to aforementioned source of uncertainty, capacity constraints might be violated at any time for a given *a priori* route. More specifically, two types of capacity violations might occur. In case *(i)*, at an origin or a transfer node, there might be an insufficient capacity on a PD vehicle to perform the pickup operation. In this case, it is assumed that the to-be-picked-up request will be served by an outsourced service at a certain cost, which is dependent on the direct distance from the point of failure to its destination. In case *(ii)*, there might be an insufficient capacity on a SL service. In this case, the excessive demand can be transported by using the subsequent service of the SL. If time windows are violated or the capacity of the subsequent SL service is exceeded, the service is outsourced similarly to case *(i)*. Below we introduce the notations and assumptions used throughout the paper.

- *Request:* Every request is associated with an origin node $r \in \mathcal{P}$ and a destination node $r+n \in \mathcal{D}$, where $n = |\mathcal{P}|$ (the number of requests to be served). Each request has one time window for the pickup node ($[l_r, u_r]$), and one time window for the delivery node ($[l_{r+n}, u_{r+n}]$). Furthermore, each freight request has demand quantity d_r that is defined with a probability distribution function. We assume that each demand unit corresponds to a package of a standardized small size (disregarding the nature of its content, e.g., dry or frozen, and its weight). In addition, we refer to a request by its corresponding pickup node.
- *Vehicle:* The set of PD vehicles is denoted by $\mathcal{V}$. In addition, each vehicle $v \in \mathcal{V}$ is associated with a capacity Q_v (i.e., maximum number of packages), the depot $o_v \in \mathcal{O}$, where $\mathcal{O}$ is the set of depots, and its corresponding time window $[l_{o_v}, u_{o_v}]$. Finally, each vehicle is assigned a routing cost per one time unit θ_v.
- *Travel and service time:* Travel and service times are known beforehand and remain unchanged during the planning horizon. Each arc (i, j) is associated with travel time Υ_{ij}. The service time at node i is represented by s_i.
- *Scheduled line:* The set of all physical transfer nodes is given as $\mathcal{S}$. The set of all physical scheduled lines is denoted by $\mathcal{E}$, which is defined by a directed arc between the start and the end of the line (i, j), where $i, j \in \mathcal{S}$. For example, between two transfer nodes i and j, two scheduled lines with opposite directions are considered as (i, j) and (j, i). $\mathcal{K}^{ij}$ represents a set of indices for the departure times from origin node i of SL (i, j), such that the departure time is denoted by p^w_{ij} (e.g., $p^0_{T_1,T_2} = 30$ units). Furthermore, the capacity of the SLs is denoted by Q_{ij}, $\forall\ (i, j) \in \mathcal{E}$. Note that scheduled lines are considered to be heterogeneous in terms of frequency and capacity. Shipping one unit of package on SL (i, j) costs η_{ij}.

Additional practical assumptions related to SLs are given as follows.

- Every SL vehicle is assumed to have a separate compartment for carrying packages, thus the passenger capacity is not influenced.
- A storage space (DHL-Packstation, 2016) for to-be-shipped packages is always available at transfer nodes. We also assume that transfers are only allowed only at the end-of-line stations in order to avoid long delays (i.e., loading/unloading) at

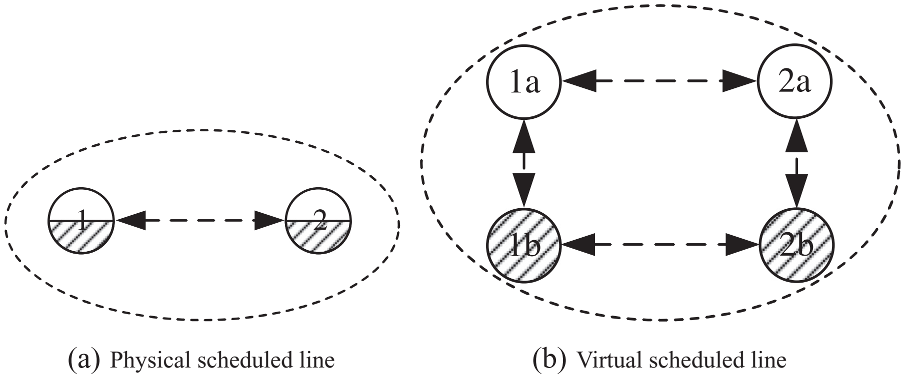

Fig. 1. An illustration of a replicated scheduled line (Ghilas et al., 2016c).

the intermediate stops. Furthermore, we assume that multiple PD vehicles can be serviced at a time at every transfer node.

- In case of multiple freight carriers using SLs, it is assumed that each carrier has a limited storage space at a transfer node and on SLs (e.g., contract-based agreement).
- Cost η_{ij} includes transportation, handling (transshipment) and storage costs.
- Since the focus of the present problem is on operational decisions, investment costs, such as the re-design of the SL vehicles and the storage capacities at the transfer nodes are disregarded.

For modeling purposes, every physical SL is assigned n copies as in Häll et al. (2009). Fig. 1a illustrates an example with one physical SL (i.e., arcs (1, 2) and (2, 1)) and two requests. Each replication is assigned to one request, and only that specific request can travel on the assigned SL (see Fig. 1b). This is done to reduce the number of decision variables in the formulation. Therefore, a set of all replicated scheduled lines is denoted by $\mathcal{F}$ (e.g., $(1a, 2a)$, $(1b, 2b)$, $(2a, 1a)$ and $(2b, 1b)$ in Fig. 1b). The set of replicated SLs associated with request r is given as $\mathcal{F}^r$ (e.g., in Fig. 1, $\mathcal{F}^a = \{(1a, 2a), (2a, 1a)\}$). In addition, the set of replicated SLs related to the replicated transfer node t is given as $\mathcal{F}^t$ (e.g., $\mathcal{F}^{1a} = \{(1\text{a}, 2\text{a}), (2\text{a}, 1\text{a})\}$ in Fig. 1). Finally, $\mathcal{F}^{ij}$ includes all replicated SLs associated with a physical SL $(i, j) \in \mathcal{E}$ (e.g., $\mathcal{F}^{1,2} = \{(1\text{a}, 2\text{a}), (1\text{b}, 2\text{b})\}$, $\mathcal{F}^{2,1} = \{(2\text{a}, 1\text{a}), (2\text{b}, 1\text{b})\}$).

Furthermore, each transfer node in $\mathcal{S}$ (e.g., nodes 1 and 2 in Fig. 1a) is copied n times. Hence, $\mathcal{T}$ represents the set of all replicated transfer nodes (e.g., $\mathcal{T} \equiv \{1a, 1b, 2a, 2b\}$ in Fig. 1b). For the sake of modeling efficiency, we use ψ^t, $\forall\, t \in \mathcal{T}$ as the physical transfer node represented by the replicated transfer node t (e.g., $\psi^{1a} = \psi^{1b} = 1$). Therefore, set $\mathcal{T}^t$ is $\{i \in \mathcal{T} |\ \psi^i = \psi^t \ and\ i \neq t\}$, $\forall\, t \in \mathcal{T}$ (e.g., $\mathcal{T}^{2a} = \{2b\}$ in Fig. 1b). Finally, let f_i^r be a constant taking value 1 if node i is the pickup node of request r, value −1 if it is the delivery node of r, or value 0 if it is none of them (pickup nor delivery node of r).

The proposed formulation is not limited to use of only one scheduled line. Any complex network topology (i.e., square, triangle, star, etc.) might be considered, and the requests are allowed to be transferred from one SL to another.

Finally, the PDPTW-SLSD can be defined on a digraph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes (i.e, $\mathcal{O} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}$) and the set of arcs $\mathcal{A} \equiv \mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 \cup \mathcal{A}_4$ defined as follows:

$$
\begin{aligned}
\mathcal{A}_1 &= ((\mathcal{P} \cup \mathcal{D})) \times (\mathcal{P} \cup \mathcal{D})) \setminus \{(r+n, r) : r \in \mathcal{P}\} \\
\mathcal{A}_2 &= \{(i, j) : i \in \mathcal{O}, j \in \mathcal{P}\} \cup \{(i, j) : i \in \mathcal{D}, j \in \mathcal{O}\} \cup (\mathcal{O} \times \mathcal{T}) \\
\mathcal{A}_3 &= ((\mathcal{P} \cup \mathcal{D})) \times \mathcal{T}) \setminus (\{(j, r) : r \in \mathcal{P}, j \in \mathcal{T}^r\} \cup \{(r+n, j) : r \in \mathcal{P}, j \in \mathcal{T}^r\}) \\
\mathcal{A}_4 &= \{(i, j) : i, j \in \mathcal{T}, (\psi^i, \psi^j) \notin \mathcal{E}\}.
\end{aligned}
$$

These sets of arcs consider all feasible connections in terms of precedence constraints. Fig. 2 presents a small example of a network with one depot, two requests and one SL service along with all possible arcs. Sets $\mathcal{A}_1$ and $\mathcal{A}_2$ are shown in Fig. 2a and sets $\mathcal{A}_3$ and $\mathcal{A}_4$ are presented in Fig. 2b.

The binary variable x_{ij}^v equals to 1 if arc (i, j) is used by PD vehicle v, 0 otherwise, $\forall\, (i, j) \in \mathcal{A}$, $v \in \mathcal{V}$. Variable y_{ij}^r takes value 1 if arc (i, j) is used by request r, 0 otherwise, $\forall\, i, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}$, $r \in \mathcal{P}$. In addition, q_{ij}^{rw} takes value 1 if replicated SL (i, j) is used by request r that departs from i at time p_{ij}^w, 0 otherwise, $\forall\, r \in \mathcal{P}$, $(i, j) \in \mathcal{F}^r$, $w \in \mathcal{K}^{\psi^i \psi^j}$. Timing decisions are assured by: α_v that indicates the time at which vehicle v arrived back to the depot, $\forall\, v \in \mathcal{V}$, by β_i that shows the time of departure of a PD vehicle from node i, $\forall\, i \in \mathcal{N}$, and finally, γ_i^r that indicates the time at which request r departed from node i, $\forall\, i \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}$, $r \in \mathcal{P}$.

The two-stage stochastic PDPTW-SLSD formulation is shown in the following two subsections.

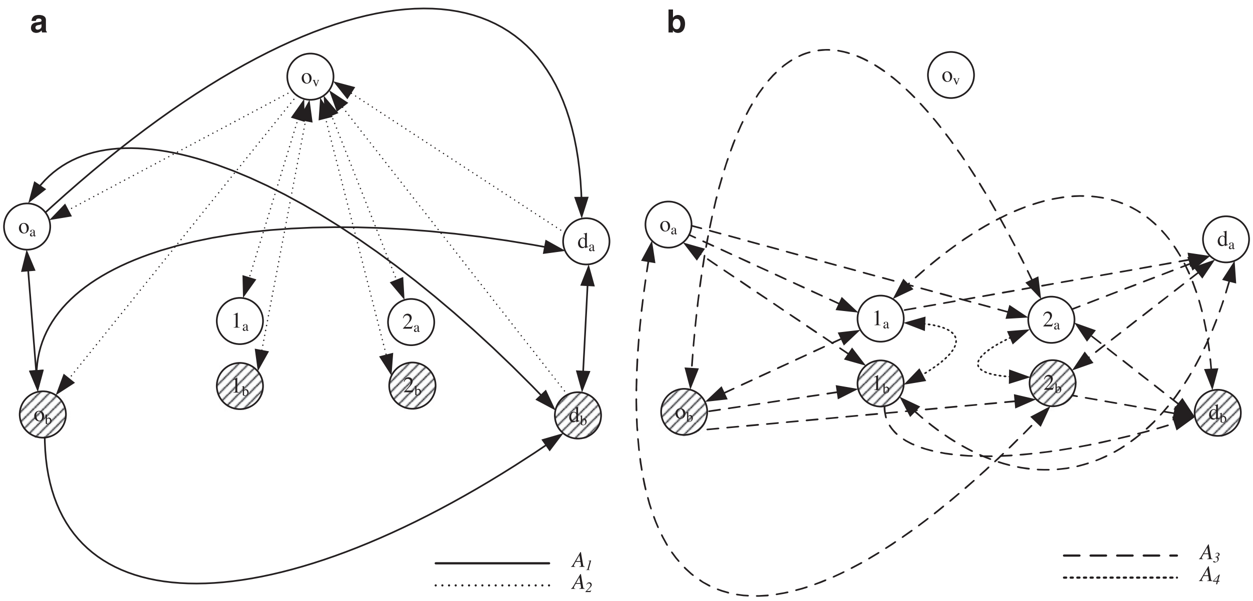

Fig. 2. An example network with two requests and one SL service (Ghilas et al., 2016a).

3.1. The first-stage model

$$\text{Minimize} \quad \sum_{(i,j)\in\mathcal{A}} \sum_{v\in\mathcal{V}} \theta_v \Upsilon_{ij} x_{ij}^v \tag{1}$$

$$+E[Q(\boldsymbol{x}, \xi, \eta)] \tag{2}$$

The objective function minimizes the total operating costs incurred by PD vehicles (1) and the *recourse* costs (2). In the recourse function, $\boldsymbol{x}$ is the given routing vector, ξ is the set of scenarios, and η is the cost vector for using SLs per unit shipped.

subject to

Routing and flow constraints

$$\sum_{i\in\mathcal{N}} \sum_{v\in\mathcal{V}} x_{ij}^v = 1 \qquad \forall j \in \mathcal{P} \cup \mathcal{D} \tag{3}$$

$$\sum_{i\in\mathcal{P}\cup\mathcal{D}\cup\mathcal{T}} x_{o_v,i}^v \le 1 \qquad \forall v \in \mathcal{V} \tag{4}$$

$$\sum_{i\in\mathcal{N}} \sum_{v\in\mathcal{V}} x_{it}^v \le 1 \qquad \forall t \in \mathcal{T} \tag{5}$$

$$\sum_{j\in\mathcal{N}} x_{ij}^v - \sum_{j\in\mathcal{N}} x_{ji}^v = 0 \qquad \forall i \in \mathcal{N}, v \in \mathcal{V} \tag{6}$$

$$\sum_{j\in\mathcal{P}\cup\mathcal{D}\cup\mathcal{T}} y_{ij}^r - \sum_{j\in\mathcal{P}\cup\mathcal{D}\cup\mathcal{T}} y_{ji}^r = f_i^r \qquad \forall r \in \mathcal{P}, i \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T} \tag{7}$$

$$\sum_{i\in\mathcal{N}} \sum_{v\in\mathcal{V}} x_{it}^v \le \sum_{r\in\mathcal{P}} \sum_{(i,j)\in\mathcal{F}^t} y_{ij}^r \qquad \forall t \in \mathcal{T} \tag{8}$$

Constraints (3) state that all pickup and delivery nodes are visited exactly once. Constraints (4) ensure that each vehicle leaves its depot at most once and constraints (5) state that each replicated transfer node is visited at most once. Flow conservations for PD vehicles and requests are considered in constraints (6) and (7), respectively. Constraints (8) ensure that if a request uses a scheduled line, a PD vehicle should pick up or drop off the request at the corresponding transfer node.

Scheduling constraints

$$y_{ij}^r = 1 \Rightarrow \gamma_j^r \ge \gamma_i^r + \Upsilon_{ij} + s_j \qquad \forall r \in \mathcal{P}, i, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T} \tag{9}$$

$$\sum_{v\in\mathcal{V}} x_{ij}^v = 1 \Rightarrow \beta_j \ge \beta_i + \Upsilon_{ij} + s_j \qquad \forall i \in \mathcal{N}, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T} \tag{10}$$

$$x_{i,o_v}^v = 1 \Rightarrow \alpha_v \ge \beta_i + \Upsilon_{i,o_v} + s_{o_v} \qquad \forall i \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}, v \in \mathcal{V} \tag{11}$$

$$\beta_{r+n} \geq \beta_r + \Upsilon_{r,r+n} + s_{r+n} \qquad \forall r \in \mathcal{P} \tag{12}$$

$$l_i \leq \beta_i - s_i \leq u_i \qquad \forall i \in \mathcal{P} \cup \mathcal{D} \tag{13}$$

$$l_{g_v} \leq \alpha_v \leq u_{o_v} \qquad \forall v \in \mathcal{V} \tag{14}$$

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

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

Timing variables of each request are considered in constraints (9). Similarly, schedules for PD vehicles are considered in constraints (10) and (11). Constraints (12) define the precedence relations for each request. Constraints (13) and (14) enforce time windows restrictions. Constraints (15)–(16) state that if a request uses a scheduled line, SL vehicle must only depart at its scheduled departure time.

Synchronization constraints

$$\sum\nolimits_{j \in \mathcal{P} \cup \mathcal{D}} y_{ij}^r = 1 \Rightarrow \gamma_i^r = \beta_i \qquad \forall r \in \mathcal{P}, i \in \mathcal{T} \tag{17}$$

$$\sum\nolimits_{j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}} y_{ij}^r = 1 \Rightarrow \gamma_i^r = \beta_i \qquad \forall r \in \mathcal{P}, i \in \mathcal{P} \cup \mathcal{D} \tag{18}$$

$$\sum\nolimits_{i \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}} y_{i,r+n}^r = 1 \Rightarrow \gamma_{r+n}^r = \beta_{r+n} \qquad \forall r \in \mathcal{P} \tag{19}$$

$$y_{tj}^r = 1 \Rightarrow \gamma_t^r = \beta_t \qquad \forall r \in \mathcal{P}, t \in \mathcal{T}, j \in \mathcal{T}^t \tag{20}$$

The set of constraints (17)–(20) ensure the synchronization between requests' and vehicles' schedules.

Decision variable domains

$$x_{ij}^v \in \{0, 1\} \qquad \forall (i, j) \in \mathcal{A}, v \in \mathcal{V} \tag{21}$$

$$y_{ij}^r \in \{0, 1\} \qquad \forall i, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}, r \in \mathcal{P} \tag{22}$$

$$\alpha_v \in R^+ \qquad \forall v \in \mathcal{V} \tag{23}$$

$$\gamma_i^r \in R^+ \qquad \forall i \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}, r \in \mathcal{P} \tag{24}$$

$$\beta_i \in R^+ \qquad \forall i \in \mathcal{N} \tag{25}$$

$$q_{ij}^{rw} \in \{0, 1\} \qquad \forall r \in \mathcal{P}, (i, j) \in \mathcal{F}^r, w \in \mathcal{K}^{\psi^i \psi^j}. \tag{26}$$

We note that constraints (9)–(11), (16) and (17)–(20) are formulated as implications. Standard linearization techniques can be easily used to express them using one or two linear inequalities, as follows:

$$\gamma_j^r \geq \gamma_i^r + \Upsilon_{ij} + s_j - M_{ij}\left(1 - y_{ij}^r\right) \qquad \forall r \in \mathcal{P}, i, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T} \tag{27}$$

$$\beta_j \geq \beta_i + \Upsilon_{ij} + s_j - M_{ij}\left(1 - \sum\nolimits_{v \in \mathcal{V}} x_{ij}^v\right) \qquad \forall i \in \mathcal{N}, j \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T} \tag{28}$$

$$\alpha_v \geq \beta_i + \Upsilon_{i,o_v} + s_{o_v} - M_{i,o_v}\left(1 - x_{i,o_v}^v\right) \qquad \forall i \in \mathcal{P} \cup \mathcal{D} \cup \mathcal{T}, v \in \mathcal{V} \tag{29}$$

$$\gamma_i^r \leq p_{ij}^w + M_i\left(2 - q_{ij}^{rw} - y_{ij}^r\right) \qquad \forall r \in \mathcal{P}, (i, j) \in \mathcal{F}^r, w \in \mathcal{K}^{ij} \tag{30}$$

$$\gamma_i^r \geq p_{ij}^w - M_i\left(2 - q_{ij}^{rw} - y_{ij}^r\right) \qquad \forall r \in \mathcal{P}, (i, j) \in \mathcal{F}^r, w \in \mathcal{K}^{ij} \tag{31}$$

$$\gamma_i^r \leq \beta_i + M_i\left(1 - \sum\nolimits_{j \in \mathcal{P} \cup \mathcal{D}} y_{ij}^r\right) \qquad \forall r \in \mathcal{P}, i \in \mathcal{T} \tag{32}$$

$$\gamma_i^r \geq \beta_i - M_i\left(1 - \sum\nolimits_{j \in \mathcal{P} \cup \mathcal{D}} y_{ij}^r\right) \qquad \forall r \in \mathcal{P}, i \in \mathcal{T}. \tag{33}$$

Constraints (18)–(20) can be linearized in the same way as constraints (17) (i.e., (32) and (33)).

3.2. The second-stage decisions

Given an a priori routing solution vector $\boldsymbol{x}$, the expected costs E[Q($\boldsymbol{x}$, ξ, η)] can be computed in two steps. In Step 1, capacity violations of the PD vehicles are evaluated. As a consequence, some requests (including transferable ones) may not

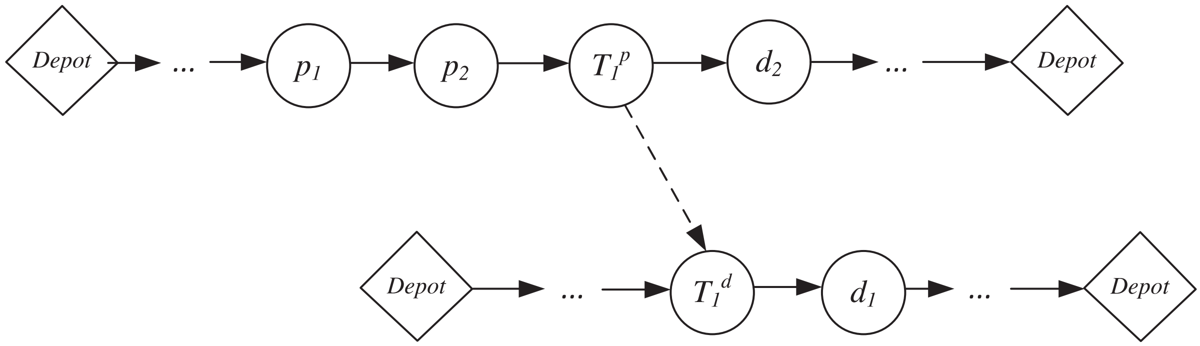

Fig. 3. An illustrative routing example of the PDPTW-SLSD.

be served by the available PD vehicles due to given demand realizations and capacity restrictions. Hence, extra costs are required for outsourcing. Further in Step 2, for remaining set of requests that are supposed to be transferred, SLs' capacity violations are verified. Each step is explained in detail below.

Step 1: For PD vehicles, capacity violations might happen either *(i)* at the pickup node of a request r (e.g., p_1 in Fig. 3), or *(ii)* at the transfer node of a request t^d, which is a destination transfer node (e.g., T_1^d in Fig. 3). In either way, it is assumed that an outsourced vehicle (e.g., taxi) could deliver the request to its destination from the point of *failure* at a cost of $P_1 \Upsilon_{l,r+n}$, $\forall\ l \in \{p_1, T_1^d\}$. Moreover, the destination node of r does not need to be visited and the shortened route leads to reduced operational costs.

Example. We now consider four possible situations: *(i)* let PD vehicle's capacity be violated at node p_2 (see Fig. 3). In this case, node d_2 will not be visited as the demand from the origin p_2 cannot be picked up by a PD vehicle. Hence, the transport service has to be outsourced. *(ii)* It is assumed that the vehicle's capacity is violated at node p_1. Hence, no related nodes to request 1 are visited: T_1^p, T_1^d, and d_1. *(iii)* Let the capacity be violated at node T_1^d. In this case, node d_1 cannot be visited. *(iv)* Let the demand of request 1 be zero. Similarly to aforementioned cases, T_1^p, T_1^d, d_1 cannot be visited. It is noted that, in cases *(i)–(iii)* the considered routes are shortened while incurring additional outsourcing costs, whereas case *(iv)* only leads to reduction in operational costs, due to lack of demand, thus no other related nodes need to be visited.

Step 2: As a result of the previous step, a restricted set of transferable requests $\overline{\mathcal{P}}^t$ is obtained. The capacity of SL service is only violated when the total demand of a set of transferable requests exceeds the capacity at a given scheduled departure time, i.e. $\sum_{r\in\overline{\mathcal{P}}^t} \sum_{(a,b)\in\mathcal{F}^{ij}} d_r q_{ab}^{rw} > k_{ij}\ \forall\ (i, j) \in \mathcal{E},\ w \in \mathcal{K}^{ij}$. In such case, the excessive demand can be stored at the origin of the SL until the next scheduled departure. If waiting leads to violation of the time window constraints, or if the SL capacity on the next trip is exceeded, then the service is outsourced as in *Step 1* (from the origin of the SL to the demand's destination). The recourse cost of a given routing solution can be calculated as shown in Algorithm 1.

The *recourse* cost variable is initialized in Line 1. For each considered scenario (Line 3), a set of routes is also stored (Line 4) and a set of transferable requests from the given routing solution is selected (Line 6). Three cases of capacity violation of a PD vehicle may exist as explained earlier. The penalty cost for not serving a request is updated in Lines 13 and 16. Once the routes are shortened, routing cost savings are updated in Line 21, where c(**X**) is the routing cost of the shortened set of routes.

The extra costs incurred by exceeding SLs capacities is updated in *getSLCost*($\mathcal{P}^t$) (Line 22). More specifically, for each SL (i, j) $\in \mathcal{E}$, $w \in \mathcal{K}^{ij}$ with violated SL capacity, the requests with the latest end time window of the corresponding delivery nodes (i.e., excess demand) are re-scheduled for the next trip on the same SL service. If time windows or SL capacity constraints are violated, the service is outsourced at an extra cost dependent on the distance to destination of the exceeded demand. As a result, the vehicle routes are shortened, i.e., the end-of-the-line station and the corresponding delivery node are not visited. Finally, line 23 updates the sample average approximation function and line 24 returns the expected *recourse* costs for a given routing solution $\boldsymbol{x}$ and a set of scenarios ξ.

3.3. An illustrative example

We now provide an example to show the necessity of using recourse actions during the transport planning. Assume an instance with two vehicles, six requests and two transfer nodes (i.e., T_1 and T_2). Nodes $p_1, \ldots, p_6$ represent pickup locations and nodes $d_1, \ldots,, d_6$ represent the corresponding delivery locations. For simplicity reasons, assume the traveling time on each arc (including SLs) to be one time unit. Two different solutions with the same routing costs are shown in Fig. 4. These solutions differ in terms of recourse costs. The numbers on top (or bottom) of each node, which are shown in italic, represent the departure times from the corresponding nodes and the numbers on top of dotted arrows indicate the request flows on the SLs.

We note that throughout the paper we refer to *stochastic* solutions as the ones that are generated using the proposed methodology in this paper, whose objective function is to minimize the routing and the recourse expected costs. On the

Algorithm 1: The generic structure of **Recourse**($\boldsymbol{x}$, ξ, η) procedure

input : A routing solution $\boldsymbol{x}$, a set of scenarios ξ
output: The average *recourse* cost $E[Q(\boldsymbol{x}, \xi, \eta)]$

Initialize $E[Q(\boldsymbol{x}, \xi, \eta)] \leftarrow 0$
Let c($\boldsymbol{x}$) be the routing costs of solution $\boldsymbol{x}$
for *scenario s in* ξ **do**
  $\mathbf{X} \leftarrow \boldsymbol{x}$
  $cost \leftarrow 0$
  $\overline{\mathcal{P}}^t \leftarrow$ set of transferable requests in $\boldsymbol{x}$
  for *each node i in* $\boldsymbol{X}$ **do**
    $r \leftarrow$ getRelatedRequest(i)
    if $i \in \mathcal{P}$ *and capacity violated* **then**
      $\mathbf{X} \leftarrow \mathbf{X} \setminus i+n$
      $\mathbf{X} \leftarrow \mathbf{X} \setminus (T_i^o \cup T_i^d)$
      $\overline{\mathcal{P}}^t \leftarrow \overline{\mathcal{P}}^t \setminus i$
      $cost \leftarrow cost + P_1 \Upsilon_{i,i+n}$
    else if *i is a destination transfer node and related request r is picked up and capacity violated* **then**
      $\mathbf{X} \leftarrow \mathbf{X} \setminus r+n$
      $cost \leftarrow cost + P_1 \Upsilon_{i,r+n}$
    else if $i \in \mathcal{P}$ *and* $d_i = 0$ **then**
      $\mathbf{X} \leftarrow \mathbf{X} \setminus i+n$
      $\mathbf{X} \leftarrow \mathbf{X} \setminus (T_i^o \cup T_i^d)$
      $\overline{\mathcal{P}}^t \leftarrow \overline{\mathcal{P}}^t \setminus i$
  $cost \leftarrow cost - c(\boldsymbol{x}) + c(\mathbf{X})$
  $cost \leftarrow cost + getSLCost(\overline{\mathcal{P}}^t, \eta)$
  set $E[Q(\boldsymbol{x}, \xi, \eta)] \leftarrow E[Q(\boldsymbol{x}, \xi, \eta)] + \frac{cost}{|\xi|}$
return $E[Q(\boldsymbol{x}, \xi, \eta)]$

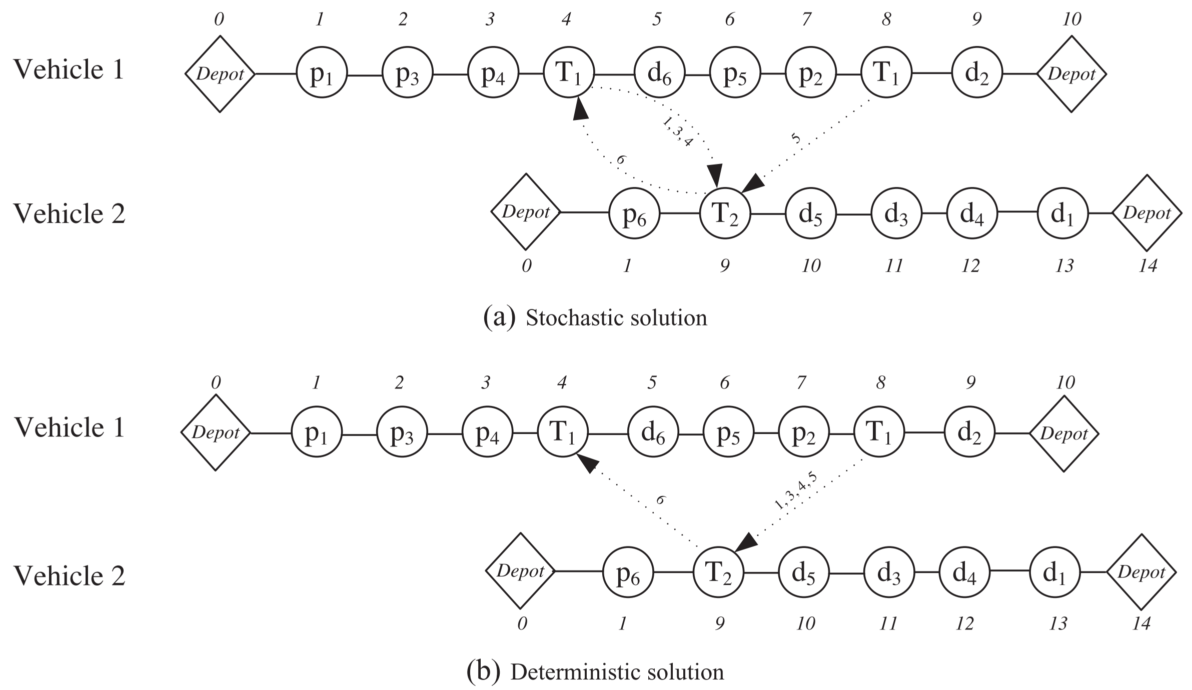

Fig. 4. Two illustrative solutions with the same routing cost.

other hand, *deterministic* solutions are generated using the methodology proposed in Ghilas et al. (2016b) (using average demands), which disregards the cost of recourse actions, thus minimizing routing and SL costs only.

In both cases, five requests, i.e., 1, 3, 4, 5 and 6, are being transported through SLs. However, the expected costs of these solutions differ. More specifically, the solution shown in Fig. 4a has a lower recourse cost as *Vehicle 1* first picks up requests 1, 3, and 4 and ships them on the SL service (i.e., T_1–T_2, see dashed arrows in Fig. 4a). Then, *Vehicle 1* picks up request 6 (which was picked up by *Vehicle 2* and shipped from T_2) from T_1, and finally requests 5 and 2 from their pickup locations. Request 5 is shipped on the SL from T_1. In this case, the total expected cost of this solution is found to be 385.40 units for a given set of scenarios.

On the other hand, in the deterministic solution (Fig. 4b), *Vehicle 1* first picks up requests 1, 3, 4 from their pickup locations and request 6 from station T_1. Then the vehicle delivers request 6 to its destination node 13 and picks up requests 5 and 2 from their pickup locations to transfer them on the SL by visiting the transfer node T_1. This solution has a total expected cost of 462.48 units and the difference from the stochastic solution is triggered by the lack of available capacity at the pickup location of request 5. Therefore, one can observe that in addition to the routing decisions, a sequence of actions (i.e., departure times on the SLs) is also important when considering the expected cost of a transport plan.

4. The ALNS-based solution methodology for the PDPTW-SLSD

In this section, we provide a solution methodology for the PDPTW-SLSD. As aforementioned, our approach consists of a scenario-based SAA framework combined with an ALNS algorithm.

4.1. The sample average approximation method

The SAA is an iterative sampling approach proposed by Kleywegt et al. (2001) to solve stochastic discrete optimization problems, where the number of scenarios is very large. Given a large scenario set Ω', which is intractable, the SAA consists of solving a set of smaller and tractable problems (referred to as SAA problems) with samples of size $|\Omega| \ll |\Omega'|$. For the PDPTW-SLSD, a sample (or a scenario) contains a demand vector for the considered requests, given probability distributions. The SAA procedure used to calculate the approximate optimality gap (or SAA gap) for the PDPTW-SLSD is as follows:

1. The algorithm takes the following input: *(i)* a sample size $|\Omega|$ to generate the SAA solution, *(ii)* a larger set of scenarios Ω' to approximate the value of the optimal solution of the SAA problem, and *(iii)* the number of iterations M.
2. For $m = 1 \rightarrow M$, do the following:
   2.1 Generate a sample set Ω and solve the corresponding SAA problem to obtain the objective value v_Ω^m and the solution x^m by using the proposed ALNS algorithm (see Section 4.2)
   2.2 Using Algorithm 1, determine the upper bound on the SAA solution for the generated sample Ω', which is denoted by $v_{\Omega'}(x^m)$. Let $\hat{x}$ be the solution that has the best value of the upper bound $v_{\Omega'}(\hat{x})$ found after a number of iterations. The variance of this estimated upper bound can be computed as follows:
$$\sigma^2_{\Omega'}(\hat{x}) = \frac{1}{|\Omega'|(|\Omega'-1|)}\sum_{i\in\Omega'}\left(v_i(\hat{x}) - v_{\Omega'}(\hat{x})\right)^2, \tag{34}$$
   where $v_i(\hat{x})$ is the objective function value of the solution $\hat{x}$ for scenario i.
   2.3 Calculate the average of the objective function values obtained in previous iterations, denoted by $\hat{v}_\Omega^m$, and their corresponding variance as shown below.
$$\hat{v}_\Omega^m = \frac{1}{m}\sum_{i=1}^{m} v_\Omega^i, \tag{35}$$
$$\sigma^2_{\hat{v}_\Omega^m} = \frac{1}{m(m-1)}\sum_{i=1}^{m}\left(v_\Omega^i - \hat{v}_\Omega^m\right)^2. \tag{36}$$
   The value $\hat{v}_\Omega^m$ is a statistical lower bound for the optimal solution value of the sample Ω' as given in Kleywegt et al. (2001). To the best of our knowledge, this is the only way that was used in the literature of approximating the statistical lower bound (see, e.g., Verweij et al., (2003); Adulyasak et al., (2015); Demir et al., (2015a)) in SAA.
   2.4 Calculate the SAA gap $\epsilon^{SAA}(\Omega, \Omega')$ and the variance of the gap as follows:
$$\epsilon^{SAA}(\Omega, \Omega') = v_{\Omega'}(\hat{x}) - \hat{v}_\Omega^m \tag{37}$$
$$\sigma^2_{\epsilon^{SAA}(\Omega,\Omega')} = \sigma^2_{\Omega'}(\hat{x}) + \sigma^2_{\hat{v}_\Omega^m}. \tag{38}$$
   The best value of $\epsilon^{SAA}(\Omega, \Omega')$ and its corresponding $\sigma^2_{\epsilon^{SAA}(\Omega,\Omega')}$ are stored during the process.
3. Return the best solution $\hat{x}$.

In the next subsection, we provide details of the ALNS heuristic algorithm, which is used to generate routing solutions of minimum total expected cost. Note that Algorithm 1 is used within ALNS to compute the recourse cost of a considered solution.

4.2. The ALNS heuristic

The ALNS heuristic was first used in Røpke and Pisinger (2006); Pisinger and Røpke (2007) to solve several variants of the VRPs. The ALNS iteratively applies a set of removal and insertion operators to modify the solution. The operators are dynamically selected according to their past and current performances. The general framework of the ALNS is given in Algorithm 2.

Algorithm 2: The ALNS framework with simulated annealing.

---

input : Removal operators D, insertion operators I, temperature T
output: X_{best}

Generate an initial solution $X_{current}$, *initialize* $X_{best} \leftarrow X_{current}$
for *a number of iterations* **do**
  Select a removal operator $d^* \in D$ *with probability* P^t_d *and apply it to* $X_{current}$ *to get the partially destroyed solution* X^d
  Select an insertion operator $i^* \in I$ *with probability* P^t_i *and apply it to* X^d *to get a new solution* X_{new}
  if $c(X_{new}) < c(X_{current})$ **then**
    $X_{current} \leftarrow X_{new}$
    if $c(X_{current}) < c(X_{best})$ **then**
      $X_{best} \leftarrow X_{current}$
  else
    if *a random number* $\epsilon \in [0, 1] < e^{-(c(X_{new})-c(X_{current}))/T}$ **then**
      $X_{current} \leftarrow X_{new}$
  Update operator probabilities and temperature T

Given an initial solution $X_{current}$, a number of requests are removed from the solution. Afterwards, these requests are re-inserted back to the partially-destroyed solution using a chosen insertion operator. A new solution X_{new} is obtained and evaluated. Three possibilities exist: *(i)* if the new solution is better than $X_{current}$, then it is accepted; *(ii)* if the new solution is worse than X_{new}, but the simulated annealing criterion is satisfied, then it is accepted; finally, *(iii)* the solution is rejected. At every iteration, the best solution X_{best} found so far is updated. The probabilities of the operators and the temperature T are updated.

The ALNS algorithm for the deterministic PDPTW-SL is first proposed by Ghilas et al. (2016b). The authors proposed effective neighborhood operators to investigate the solution space and an efficient feasibility-check mechanism. In this paper, we extend the ALNS proposed by Ghilas et al. (2016b) to solve the SAA problems of the PDPTW-SLSD as discussed in Section 4.1. The performance of the algorithm is enhanced by adapting several operators in stochastic setting. The details of the algorithm used are given in the following subsections.

4.2.1. Initial solution

To obtain an initial solution to the problem, a greedy insertion heuristic is used. For every vehicle an empty route is constructed with the appropriate capacity level and depot. The requests are stored in a list $\mathcal{L}$ and are greedily inserted one by one in a random order (greedy insertion). The objective is to minimize (1) and (2), namely routing and recourse costs (see Algorithm 1) simultaneously. At this stage, we assure the feasibility in terms of SL schedule and time windows.

4.2.2. Removal operators

We use ten removal operators in the proposed ALNS algorithm. As aforementioned, the aim of the removal stage is to partially destroy a given solution X, by removing ϕ requests and adding them to a *removal list* $\mathcal{L}$. All ten operators were inspired or adapted from Røpke and Pisinger (2006); Demir et al. (2012) and Ghilas et al. (2016b). We first present the so-called *stochastic* removal operators.

- *Shaw removal (SR)*: The SR operator removes a number of requests that are related in a predefined way and therefore are easy to change. The procedure first randomly selects one request r_1 and then adds it to the removal list $\mathcal{L}$. Let $l_{r_1,r_2} = -1$ if two nodes, one related to r_1 (i.e., r_1 or $r_1 + n$) and another to r_2 (i.e., r_2 or $r_2 + n$) are in the same route, and 1 otherwise. SR selects the request $r^* = argmin_{r_2 \in \mathcal{P}}\{\Pi_1[\Upsilon_{r_1,r_2} + \Upsilon_{r_1+n,r_2+n}] + \Pi_2[|\beta_{r_1} - \beta_{r_2}| + |\beta_{r_1+n} - \beta_{r_2+n}|] + \Pi_3 l_{r_1,r_2} + \Pi_4(|\bar{d}_{r_1} - \bar{d}_{r_2}| + |\sigma^2_{r_1} - \sigma^2_{r_2}|)\}$, where Π_1–Π_4 are the normalize weights, $\bar{d}_{r_1}$ and $\sigma^2_{r_1}$ are the mean of the demand of request r_1 given the probability distribution and its variance, respectively.
- *Demand-based removal (DR)*: DR is a special case of SR with $\Pi_4 = 1$, and $\Pi_1 = \Pi_2 = \Pi_3 = 0$.
- *Worst expected removal (WER)*: This operator removes a number of requests with the highest total expected costs. The cost is determined as follows: for a given routing solution, the cost of request r is the difference in the objective function between two solutions, namely one where request r is served and the solution were request r is not served.

We now briefly present so-called *deterministic* removal operators. For more details, the reader is referred to Ghilas et al. (2016b).

- *Random removal (RR)*: This operator randomly removes a number of requests from the solution. Random selection helps diversifying the search.
- *Route removal (ROR)*: This operator removes a randomly selected route from the solution. The ROR sequentially removes a node j from the selected route until all nodes are removed. The related nodes of j, i.e., its pickup, delivery node, or transfer nodes are also removed.
- *Worst-distance removal (WDR)*: This operator removes a number of requests, that require long detours to visit their corresponding pickup and delivery nodes in the given routes. In other words, this operator iteratively removes request $r^* = argmax_{r\in\mathcal{P}}\{|\Upsilon_{i-1,r} + \Upsilon_{r,i+1} + \Upsilon_{j-1,r+n} + \Upsilon_{r+n,j+1}|\}$, where $i-1$ and $j-1$, are predecessors and $i+1$ and $j+1$ successors of the pickup and delivery nodes, respectively.
- *Late-arrival removal (LAR)*: Operator LAR removes a number of requests with the largest deviation of the service start time from l_r and l_{r+n}, respectively. In other words, the procedure iteratively removes a request $r^* = argmax_{r\in\mathcal{P}}\{|[\beta_r - s_r - l_r] + [\beta_{r+n} - s_{r+n} - l_{r+n}]|\}$, where β_i is the departure time from node i. The idea is to avoid delayed service start times or long waiting times.
- *Proximity-based removal (PR)*: This operator is a special case of SR operator with $\Pi_1 = 1$, and $\Pi_2 = \Pi_3 = \Pi_4 = 0$.
- *Time-based removal (TR)*: The operator is a special case of the SR with $\Pi_2 = 1$, and $\Pi_1 = \Pi_3 = \Pi_4 = 0$.
- *Historical knowledge node removal (HR)*: This neighborhood operator stores the position cost of every request r. This cost is calculated as $c_r = \Upsilon_{i-1,r} + \Upsilon_{r,i+1} + \Upsilon_{j-1,r+n} + \Upsilon_{r+n,j+1}$ at every iteration. The best position cost c_r^* of request r is updated to be the minimum of all c_r values calculated. The HR operator then selects the request r^* with the maximum deviation from its best position cost, i.e., $r^* = argmax_{r\in\mathcal{P}}\{c_r - c_r^*\}$ and adds it to $\mathcal{L}$.

4.2.3. Insertion operators

The insertion operators repair a partially-destroyed solution by re-inserting the requests from the removal list $\mathcal{L}$ back into the routes. Five construction heuristics are used as insertion operators and these are described as follows:

- *Greedy insertion (GI)*: The GI repeatedly inserts a request (both pickup and delivery nodes) in the cheapest possible way. The requests are chosen randomly from $\mathcal{L}$. Let $\Delta^r_{I_iJ_j}$ be the objective function value when the pickup node of r is inserted in route I (position i) and its corresponding delivery node is inserted in route J (position j). Thus, $\Delta^r_{I_iJ_j}* = argmin\{\Delta^r_{I_iJ_j}\}$.
- *Greedy insertion with noise function (GIN)*: GIN is an extension of GI with a modified cost function $NewCost = ActualCost + \bar{d}\,\mu\,\epsilon$, where $\bar{d}$ is the longest distance between nodes, μ is a noise parameter used to diversify the cost function and is set equal to 0.1, and ϵ is a random number between $[-1, 1]$.
- *Second-best insertion (SI)*: This operator selects the second best insertion for all considered requests. The SI helps diversify the search.
- *Second-best insertion with noise function (SIN)*: This operator is an extension of SI and uses the same cost function as the GIN operator.
- *Best out of λ feasible insertions (λFI)*: For each selected request, this operator selects the best option found after first λ feasible insertions. The constant λ is a random number $\in [1, \psi]$. For each request $r \in \mathcal{L}$, this operator generates a set of possible insertions by assuring the precedence constraints. This set is then sorted in increasing order based on the insertion cost (distance-based) and the search starts following the first cheapest insertion criterion.

Furthermore, five variants of the aforementioned operators are used, where $\mathcal{L}$ is a sorted list, in terms of time flexibility (i.e., the least flexible first) of requests. The flexibility of request r is calculated as $|u_{r+n} - l_r|$. We note that unlike classical pickup and delivery problem, the PDPTW-SLSD may have the pickup and delivery nodes of certain requests positioned in different routes. This requires a solution to include at least one SL. To handle this, the procedure introduced in Ghilas et al. (2016b) is used to insert transfer nodes by considering the precedence constraints. In addition to aforementioned, checking the feasibility with respect to time windows is not straightforward and therefore we use the procedure introduced in Ghilas et al. (2016b) to do that. Finally, in order to avoid redundant calculations, time windows are used to check for feasible positions as described in Braekers et al. (2014).

4.2.4. Adaptive score adjustment procedure

A roulette-wheel mechanism controls the selection of the removal and insertion operators. Initially, each removal and insertion operators are assigned a *probability* (i.e., the probability of each is set to 0.1). After N_w iterations, the probabilities are updated using the following formula: $P_d^{t+1} = P_d^t\,(1 - r_p) + r_p\,\pi_i/\omega_i$, where r_p is the roulette-wheel constant, π_i and ω_i are the score and the number of times operator i was used during the last N_w iterations, respectively. The score of operators measures their effect on the search progress. If a new best solution is found, the score of the removal and insertion operators is increased by σ_1. If the new solution outperforms the current solution, the score is increased by σ_2. If the solution quality deteriorates but the solution is accepted, the score is increased by σ_3.

5. Computational experiments

This section presents the results of computational experiments performed to assess the performance of the proposed methodology. We first describe the instance characteristics and the parameters used.

5.1. Data and experimental setting

Three sets of instances *R, C*, and *RC* were adapted from Ghilas et al. (2016a) to consider stochastic demands. Each instance contains up to three SLs in a triangular topology. The SLs have a frequency of one departure every 30 time units. Each instance contains up to 40 requests (i.e., up to 40 pick-up and 40 delivery nodes) over 200 × 200 Euclidean space. Instances are named as $G\_n\_sl\_v$, where *G* is the geographic distribution of the customers, *n* is the number of to-be-served requests, *sl* is the number of available SLs, and *v* is the number of available PD vehicles. By geographic distribution *G* we mean the following: *C* – clustered request nodes around transfer nodes, *R* – uniform-randomly distributed request nodes, and finally *RC* – randomly clustered nodes. More specifically, *C* has nodes positioned within at most 30 time units to one of the three available transfer nodes whereas *RC* has nodes within 80 time units respectively. In all cases, two depots with up to six PD vehicles each are considered.

The planning horizon is set to 600 time units. The time windows are randomly generated with an average width of 40 units. Service times are set to at most three time units. The request demands are assumed to follow a discrete triangular distribution for a given minimum value *a*, mean *b* and maximum value *c*. The capacity of each PD vehicle is generated between six and nine units. The carrying capacity on the considered SLs is assumed to be five demand units. The instances can be found on the web page of SmartLogisticsLab (2016).

The parameters used in the computational experiments are shown in Table 1. Heuristic-specific parameters have been assigned the same values as in Ghilas et al. (2016b), which were used to solve the deterministic PDPTW-SL. The settings used generally worked well on the test problems with uncertain demands. We assume a driving time unit cost for PD vehicles to be 0.5 units. It seems quite reasonable when considering the driver's wage, fuel consumption, insurance, taxes, and etc. Furthermore, the cost of using SL service is set to 1 unit and it includes transportation, handling and storage costs. The recourse cost for outsourcing the service, which is considered per driving time unit, is assumed to be 3 units.

5.2. The performance of the operators

To show the number of times each removal and insertion operator was called within the ALNS, we give relevant information in Tables 2 and 3. These tables indicate the usage frequency as a percentage of the total number of iterations, and the total time spent (shown in parenthesis) of each operator, respectively. We note that the results are obtained using only three instances of different sizes.

Table 1
Parameters used in the methodology.

Notation	Definition	Value	Notation	Definition	Value
θ_v	Cost of vehicle *v*	0.5	$\|\Omega'\|$	The size of the large set of scenarios	10,000
η_{ij}	Cost of SL (*i, j*)	1	$\|\Omega\|$	The size of the small set of scenarios	60
P_1	Cost for outsourcing	3	M	Number of SAA iterations	10
k_{ij}	Capacity of SL (*i, j*)	5	σ_1	Score for new best solution	1
	Number of ALNS interations	10,000	σ_2	Score for improving the current solution	3
λ	Number of feasible insertions (λFI)	30	σ_3	Score for accepting a worse solution	5

Table 2
Number of iterations as a percentage of the total number iterations and the CPU times required by the removal operators.

Instance	RR	ROR	LAR	WDR	SR	PR	DR	TR	HR	WER
C10_1_4	7.89 (0.00)	**10.52 (0.00)**	**21.05 (0.01)**	5.26 (0.00)	5.26 (0.00)	5.26 (0.00)	7.89 (0.00)	5.26 (0.00)	**10.52 (0.00)**	**21.05 (0.01)**
RC15_1_6	3.44 (0.00)	6.89 (0.00)	1.72 (0.00)	**22.41 (0.01)**	8.62 (0.01)	6.89 (0.00)	5.17 (0.00)	8.62 (0.01)	**24.16 (0.01)**	**12.08 (0.01)**
R20_1_8	1.72 (0.00)	3.12 (0.00)	10.93 (0.01)	**12.49 (0.01)**	1.56 (0.00)	7.81 (0.01)	**12.49 (0.01)**	4.68 (0.00)	**14.06 (0.01)**	**31.14 (0.02)**

Table 3
Number of iterations as a percentage of the total number iterations and the CPU times required by the insertion operators.

Instance	GI	SI	GIN	SIN	λFI	oGI	oSI	oGIN	oSIN	oλFI
C10_1_4	5.89 (0.89)	7.62 (1.15)	8.22 (1.24)	3.64 (0.92)	3.64 (0.55)	**22.07 (3.33)**	**17.25 (2.30)**	**17.11 (3.68)**	3.66 (0.90)	10.90 (1.94)
RC15_1_6	2.86 (0.36)	6.13 (0.78)	6.26 (0.80)	1.64 (0.58)	1.85 (0.23)	**26.29 (3.37)**	**22.24 (2.85)**	**19.04 (2.54)**	1.18 (0.57)	12.51 (1.86)
R20_1_8	7.89 (2.83)	7.29 (2.51)	9.38 (2.99)	1.34 (0.51)	3.03 (0.97)	**23.26 (7.43)**	**24.06 (7.68)**	**16.94 (5.41)**	1.27 (0.59)	5.54 (2.09)

Table 4
Comparison between the PDPTW-SD and the PDPTW-SLSD with different number of SLs.

Instance	PDPTW-SD		SL = 1			SL = 2			SL = 3			CPU
	E(x)	#vehs	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs	(s)
C10_SL_4	966.98	3	931.65	4	4	924.55	4	8	860.94	4	12	378
C15_SL_6	1440.96	4	1363.13	5	9	1336.68	4	19	1290.08	6	23	863
C20_SL_8	1291.81	5	1229.41	6	6	1223.27	7	12	1163.13	7	18	1582
C25_SL_8	1742.78	6	1538.03	7	13	1536.38	7	13	1449.71	7	22	2111
C40_SL_8	1978.31	8	1930.53	8	9	1906.94	7	21	1851.45	8	20	2532
RC10_SL_4	1105.20	4	988.49	3	3	987.90	3	3	954.52	4	5	403
RC15_SL_6	1614.19	6	1614.42	6	0	1605.58	6	1	1580.85	6	2	418
RC20_SL_8	2038.22	7	1815.02	7	6	1779.64	6	12	1746.79	7	10	1144
RC25_SL_8	2196.56	8	2179.62	8	5	2057.69	8	14	2000.96	8	14	1823
RC40_SL_12	2940.16	11	2898.77	10	2	2892.61	11	7	2796.76	11	8	2753
R10_SL_4	874.60	4	873.01	4	2	871.18	4	2	870.54	4	4	209
R15_SL_6	1535.15	6	1530.84	6	0	1513.99	6	1	1494.42	6	6	475
R20_SL_8	2179.94	8	2179.58	8	0	2119.65	8	3	2087.96	8	3	939
R25_SL_8	2656.76	8	2583.78	8	4	2549.66	8	6	2549.51	8	6	1878
R40_SL_12	3651.34	10	3633.18	11	3	3568.82	10	7	3555.55	11	6	2579
Average	1880.86	7	1819.30	7	4	1791.64	7	9	1750.21	7	11	1339

Table 2 shows that WDR, HR and WER are the most frequently used operators. The first operator is used to consider traveling distance aspect of the problem whereas the HR and WER operators consider the objective function value (including uncertainty aspect) of the PDPTW-SLSD.

Table 3 shows that the most frequently used insertion operators are oGI, oSI and oGIN. Ordered insertion operators tend to perform better than randomly-ordered ones. Furthermore, insertion operators with noise functions are used less than the operators without noise function.

5.3. Comparison between the PDPTW-SD and the PDPTW-SLSD

In this section, we identify the effects of using SLs on the total operational costs (including expected recourse costs). Table 4 shows the obtained results for four different scenarios: *(i)* the classical PDPTW with stochastic demands, where no transfers are allowed, *(ii)*–*(iv)* the PDPTW-SLSD with one, two and three available SLs. The results indicate the average value over the 10 runs of the algorithm. Column *E(x)* indicates the total expected cost, *#vehs* shows the number of vehicles used, and finally *#SLs* indicates the number of demand units (i.e., mean value of the given probability distribution of each request) that used SL services. Finally, the average CPU time needed to solve the PDPTW-SLSD is shown in the last column.

According to the obtained results, the flexibility of using SLs as a part of transport plan leads to reasonable amount of operational cost savings. More specifically, on average, the total expected costs can be decreased by 7%. In some cases (e.g., RC20_3_8) using SLs may even lead to 14% cost savings. However, in some cases (e.g., RC15_1_8), the integrated networks do not lead to any benefits as SLs are not used. There may be several reasons for not using the available SLs. For example: time windows of the requests may not have sufficient slack time for using SLs, or the pickup and delivery nodes are close to the same transfer node. Fig. 5 illustrates the average cost savings by considering all three scenarios (i.e., 1-, 2-, and 3-SLs) on three instance sets (*C*, *RC*, and *R*).

As can be observed from Table 4 and Fig. 5, if more request nodes are clustered around the transfer nodes, larger operational cost savings can be achieved. More specifically, integrated networks yield larger benefits in *C* and *RC* instances (i.e., up to 16% and 14%, respectively), compared to *R* instances (i.e., up to 4%). In addition, the more SLs are available, the larger integration between the transportation networks is, hence larger cost savings can be achieved. In other words, all instance sets with one available SL lead to more costly solutions, compared to the 2-SLs and 3-SLs cases. It can be observed in Table 4 that the more SLs are available, the more freight units are served using public transportation services, thus decreasing the expected costs.

5.4. The relationship between SL capacity and total expected costs

In this section, we investigate the effect of SLs capacity on the total expected costs. We test three scenarios with different SLs capacity: *five*, 10, 15 demand units. The average results out of ten runs of the algorithm are shown in Table 5.

On average, one can observe a trend (Fig. 6), which indicates that larger SL service capacity lead to fewer violations, hence fewer recourse actions needed to tackle SL service capacity violations. With an extra capacity, the PDPTW-SLSD solutions become cheaper (up to 0.44%) and slightly more freight units are shipped via SLs.

It is important to note that, in some cases (e.g., RC15_1_6), statistically insignificant difference in terms of total expected cost can be observed. This is explained by the fact that each solution is evaluated on different sets of independently generated scenarios for a given demand distribution.

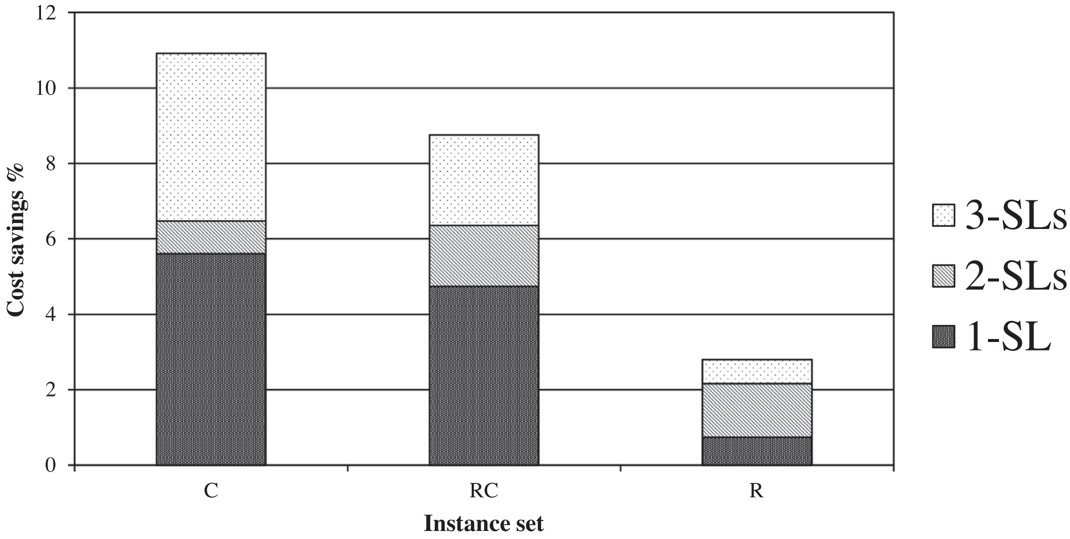

Fig. 5. Average savings by using different number of SLs.

Table 5
The effect of SLs capacity on the total expected operational costs.

	$k_{ij} = 5$			$k_{ij} = 10$			$k_{ij} = 15$		
Instance	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs
C10_1_4	931.65	4	4	926.23	4	5	926.23	4	5
C15_1_6	1363.13	5	9	1361.08	5	9	1361.89	5	9
C20_1_8	1229.41	6	6	1218.96	6	6	1218.98	6	6
C25_1_8	1538.03	7	13	1522.38	7	13	1522.39	7	13
C40_1_8	1930.53	8	9	1894.53	8	10	1868.89	8	10
RC10_1_4	988.49	3	3	986.35	3	3	986.80	3	3
RC15_1_6	1614.42	6	0	1615.27	6	0	1615.45	6	0
RC20_1_8	1815.02	7	6	1809.18	7	6	1809.95	7	6
RC25_1_8	2179.62	8	5	2177.85	8	6	2177.95	8	6
RC40_1_12	2898.77	10	2	2898.70	11	2	2898.41	10	3
R10_1_4	873.01	4	2	862.91	4	7	862.71	4	7
R15_1_6	1530.84	6	0	1530.60	6	0	1530.63	6	0
R20_1_8	2179.58	8	0	2179.27	8	0	2177.93	8	0
R25_1_8	2583.78	8	4	2582.18	8	4	2582.44	8	4
R40_1_12	3633.18	11	3	3628.10	10	4	3628.72	11	5
Average	1819.30	7	4	1812.91	7	5	1811.29	7	5

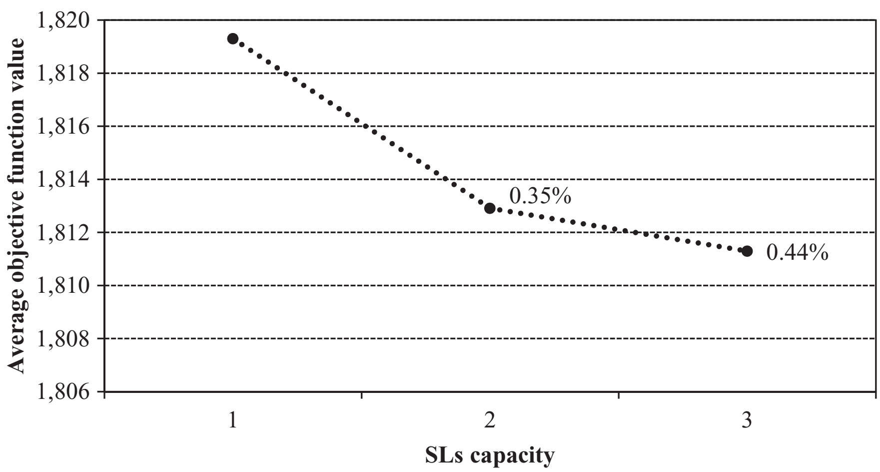

Fig. 6. The effect of SLs capacity.

Table 6
The effect of the SLs frequency on the total expected costs.

	7 units			30 units			180 units			PDPTW-SD	
Instance	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs	E(x)	#vehs
C10_1_4	905.53	4	8	931.65	4	4	932.30	4	4	966.98	3
C15_1_6	1361.64	5	9	1363.13	5	9	1374.10	4	9	1440.96	4
C20_1_8	1218.82	6	6	1229.41	6	6	1291.88	5	0	1291.81	5
C25_1_8	1527.91	7	13	1538.03	7	13	1665.44	7	5	1742.78	6
C40_1_8	1882.80	8	9	1930.53	8	9	1973.03	8	3	1978.31	8
RC10_1_4	987.22	3	3	988.49	3	3	1103.76	4	0	1105.20	4
RC15_1_6	1614.42	6	0	1614.42	6	0	1614.50	6	0	1614.19	6
RC20_1_8	1812.13	7	6	1815.02	7	6	2038.35	7	0	2038.22	7
RC25_1_8	2179.77	8	6	2179.62	8	5	2193.69	8	0	2196.56	8
RC40_1_12	2898.78	11	2	2898.77	10	2	2918.31	11	1	2940.16	11
R10_1_4	873.45	4	0	873.01	4	2	874.62	4	1	874.60	4
R15_1_6	1530.47	6	0	1530.84	6	0	1534.37	6	0	1535.15	6
R20_1_8	2176.10	8	0	2179.58	8	0	2179.13	8	0	2179.94	8
R25_1_8	2583.16	8	4	2583.78	8	4	2588.67	8	4	2656.76	8
R40_1_12	3624.64	10	3	3633.18	11	3	3635.61	10	3	3651.34	10
Average	1811.79	7	5	1819.30	7	4	1861.18	7	2	1880.86	7

Table 7
Heterogeneous PD vehicle routing costs.

Instance	PDPTW-SLSD			PDPTW-SD		
	E(x)	#vehs	#SLs	E(x)	#vehs	Savings %
C10_1_4	1061.22	4	2	1067.11	4	0.55
C15_1_6	1473.20	5	9	1630.01	4	9.62
C20_1_8	1430.77	6	6	1521.17	6	5.94
C25_1_8	1827.49	7	13	2117.86	6	13.71
C40_1_8	2423.78	8	6	2525.79	8	4.04
RC10_1_4	1050.29	3	3	1187.57	4	11.56
RC15_1_6	1975.01	6	0	1975.80	6	0.04
RC20_1_8	2222.48	7	6	2458.30	7	9.59
RC25_1_8	2750.97	8	2	2762.94	8	0.43
RC40_1_12	3671.89	10	2	3678.82	11	0.19
R10_1_4	940.25	4	4	954.99	4	1.54
R15_1_6	1648.02	6	0	1649.34	6	0.08
R20_1_8	2666.05	8	1	2689.03	8	0.85
R25_1_8	3169.48	8	4	3233.18	8	1.97
R40_1_12	3986.54	11	4	4015.53	10	0.72
Average	2153.16	7	4	2231.16	7	4.06

5.5. The SLs frequency analysis

In this section, we quantify the effect of the SLs frequency on the total expected costs of the PDPTW-SLSD. We test three situations, namely one departure every seven time units, the original case (i.e., every 30 time units) and, finally, one departure every 180 time units. The average over 10 runs of the algorithm is shown for each instance in Table 6.

According to the obtained results, lower operational costs can be achieved with higher frequency of the SLs. The cost savings increase due to the fact that more requests are shipped via SLs. On the other hand, higher frequency implies more SLs capacity, hence higher integration between the two transportation networks that leads to fewer SLs capacity violations.

5.6. An analysis on heterogeneous PD vehicle routing costs

In this section, we investigate the benefits of the integrated networks considering heterogeneous routing costs of the PD vehicles and one SL. More specifically, the minimum-capacity vehicle is assumed to cost 0.5 per operating time unit. Larger vehicles are assigned a cost that linearly increases with the carrying capacity. The average results over the 10 runs of the algorithm are presented in Table 7. The columns are self-explanatory.

The results indicate that operating cost savings of up to 13% can be achieved compared to the corresponding PDPTW-SD solutions. On average, using the same amount of vehicles as in classical pickup and delivery system, and the available SLs may lead to average cost savings of 4%.

Table 8
Sensitivity analysis on the ratio of costs.

$\theta \backslash \eta$	1			10			30			PDPTW-SD	
	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs	E(x)	#vehs	#SLs	E(x)	#vehs
0.5	406.72	6	37	419.42	4	0	419.58	4	0	419.84	4
1	790.40	5	35	823.77	6	11	829.54	4	0	829.87	4
5	3757.50	6	36	3851.50	6	32	3878.86	6	30	4199.47	4

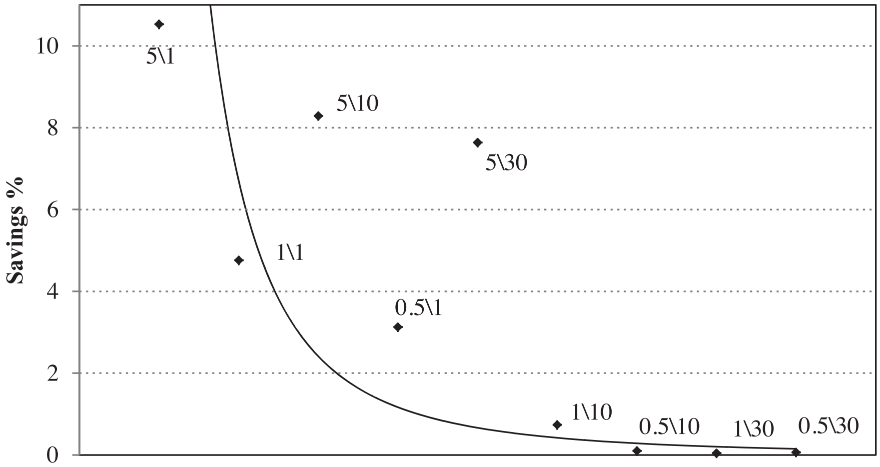

Fig. 7. Relationship between the operating cost savings and $\theta \backslash \eta$.

5.7. Case study of Amsterdam metro system

In this section, we investigate the performance of the PDPTW-SLSD on a realistic scheduled-line network. More specifically, we replicate the Amsterdam metro system as a case study. In addition, one instance with 25 requests and two depots with three vehicles each is considered. We evaluate the systems' performance on different cost (i.e., routing and SL) settings. Note that outsourcing cost P_1 is set to $6 \times \theta$, where θ is considered to be the same for all PD vehicles. Table 8 indicates the obtained results.

Fig. 7 illustrates the relationship between operational cost savings and the ratio $\theta \backslash \eta$. As can be seen, the larger routing cost is, the more savings can be achieved from the integrated transportation networks (see 5\1, 5\10, etc. in Fig. 7).

For a given SL cost η, the larger θ is, the more requests will be shipped via available SLs to perform the deliveries, and thus more cost savings can be achieved. On the other hand, smaller θ leads to insignificant incentives to make use of SLs, unless their cost is very small. Otherwise, the cost of using SLs will trade-off the routing cost savings, thus reducing the benefits of integrating system to none.

It is important to note that the challenges for integrating freight movement with public transport services include complex issues of public transportation (e.g., tight timetables), differences between customer-focused companies versus operations-focused companies (e.g., time and cost), obligations of the parties involved in each business (e.g., labor unions), availability of reliable and consistent data (e.g., lack of real-time data), and limited responses to the technological advances (e.g., design of new stations). All above cited attempts to augment public transport services with urban freight transport in real-life proved to be unsuccessful. In our view, the main reason is the lack of decision support tools that help generate reliable delivery plans. Secondly, lack of coordination and information sharing between actors involved played a crucial role. Finally, lack of sufficient financial resources needed for constructing storage facilities and re-designing the public transport vehicles had an impact on the outcome of the projects. These challenges could be reduced by increasing the communication between the parties involved and proposing new algorithms and methodologies to tackle the uncertainty of the transportation systems.

6. Conclusions

We proposed a scenario-based solution methodology to solve the considered short-haul transportation problem, where demand uncertainty is also taken into consideration. ALNS heuristic algorithm is used within the SAA framework to solve the PDPTW-SLSD. The computational results on instances with up to 40 requests (i.e., 80 locations) indicate significant reduction in expected operational costs, compared to the corresponding PDPTW stochastic routing solutions (i.e., up to

16%). In addition, using more SLs leads to shorter vehicle routes and fewer capacity violations. As a consequence of using extra capacity, lower operational costs can be achieved. Intuitively, larger SL capacity or higher SL frequency benefits the transportation system since fewer recourse actions need to be taken. Furthermore, the results indicate that the relationship between routing and SL costs drives the cost savings.

There are a number of challenges that remain to be tackled before implementing such an integrated transportation system. In particular, other sources of uncertainty (e.g., travel times) need to be taken into consideration in the decision support tools. Re-designing the public transportation (e.g., schedules, routes, storage facilities) for transporting freight is also needed to be done in a cost-efficient way. In terms of future research, investigating exact algorithms for the PDPTW-SLSD may be an interesting topic. Therefore, efficient mathematical formulation for the recourse problem is needed to apply exact algorithms based on stochastic programming techniques. Last, but no least, focusing on environmental aspects, i.e., minimizing the emissions instead of operating costs, may be an interesting research area.

Acknowledgments

The authors gratefully acknowledge funds provided by the Dutch Institute for Advanced Logistics (DINALOG), under the grant titled "Cargo Hitching", # 2011 4 086R. Thanks are due to the Editor-in-chief, Area Editor, and three anonymous reviewers for their useful comments and for raising interesting points for discussion.

References

Adulyasak, Y., Cordeau, J-F., Jans, R., 2015. Benders decomposition for production routing under demand uncertainty. Operations Research 63 (4), 851–867.
Aldaihani, M., Dessouky, M., 2003. Hybrid scheduling methods for paratransit operations. Computers and Industrial Engineering 45 (1), 75–96.
Bianchi, L., Dorigo, M., Gambardella, L., Gutjahr, W., 2009. A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing 8 (2), 239–287.
Braekers, K., Caris, A., Janssens, G., 2014. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. Transportation Research Part B: Methodological 67, 166–186.
CargoTram, 2007. Official webpage (accessed on 11.01.2016) http://www.eltis.org/discover/case-studies/delivering-goods-cargo-tram-amsterdam-netherlands.
Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. European Journal of Operational Research 223 (2), 346–359.
Demir, E., Bektaş, T., Laporte, G., 2014. A review of recent research on green road freight transportation. European Journal of Operational Research 237 (3), 775–793.
Demir, E., Burgholzer, W., Hrušovskỳ, M., Arıkan, E., Jammernegg, W., Woensel, T.V., 2015a. A green intermodal service network design problem with travel time uncertainty. Transportation Research Part B: Methodological. (in press)
Demir, E., Huang, Y., Scholts, S., Van Woensel, T., 2015b. A selected review on the negative externalities of the freight transportation: modeling and pricing. Transportation Research Part E: Logistics and Transportation Review 77, 95–114.
DHL-Packstation, 2016. DHL official webpage (accessed on 11.01.2016). URL www.dhl.de/en/paket/pakete-empfangen/packstation.html.
Drexl, M., 2012. Synchronization in vehicle routing - a survey of vrps with multiple synchronization constraints. Transportation Science 46, 297–316.
Dumas, Y., Desrosiers, J., Soumis, F., 1991. The pickup and delivery problem with time windows. European Journal of Operational Research 54 (1), 7–22.
Errico, F., Crainic, T., Malucelli, F., Nonato, M., 2013. A survey on planning semi-flexible transit systems: Methodological issues and a unifying framework. Transportation Research Part C: Emerging Technologies 36, 324–338.
Ghilas, V., Cordeau, J-F., Demir, E., Van Woensel, T., 2016a. Branch-and-Price for the Pickup and Delivery Problem with Time Windows and Scheduled Lines. Technical Report. Industrial Engineering & Innovation Sciences, Operations, Planning, Accounting, and Control, Eindhoven University of Technology.
Ghilas, V., Demir, E., Van Woensel, T., 2016b. An adaptive large neighborhood search heuristic for the pickup and delivery problem with scheduled lines. Computers & Operations Research 72, 12–30.
Ghilas, V., Demir, E., Van Woensel, T., 2016c. The pickup and delivery problem with time windows and scheduled lines. INFOR: Information Systems and Operational Research. (in press)
Gutjahr, W., 2011. Recent trends in metaheuristics for stochastic combinatorial optimization. Central European Journal of Computer Science 1, 58–66.
Häll, CH., Andersson, H., Lundgren, J., Varbrand, P., 2009. The integrated dial-a-ride problem. Public Transportation 1, 39–54.
Heilporn, G., Cordeau, J-F., Laporte, G., 2011. An integer L-shaped algorithm for the dial-a-ride problem with stochastic customer delays. Discrete Applied Mathematics 159 (9), 883–895.
Hurtigruten, 2013. Hurtigruten official webpage (accessed on 11.01.2016) URL www.hurtigruten-web.com/index_en.html.
Kleywegt, A., Shapiro, A., Homem-De-Mello, T., 2001. The sample average approximation method for stochastic discrete optimization. SIAM Journal of Optimization 2, 898–913.
Laporte, G., Louveaux, F., van Hamme, L., 2002. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. Operations Research 50 (3), 415–423.
Levin, Y., Nediak, M., Topalôglu, H., 2012. Cargo capacity management with allotments and spot market demand. Operations Research 60 (2), 351–365.
Liaw, C., White, C., Bander, J., 1996. A decision support system for the bimodal dial-a-ride problem. IEEE Transactions on Systems, Man, and Cybernetics, Part A 26 (5), 552–565.
Louveaux, F., Gonzalez, J., 2009. On the one-commodity pickup-and-delivery traveling salesman problem with stochastic demands. Mathematical Programming 119 (1), 169–194.
Lu, Q., Dessouky, M., 2006. A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. European Journal of Operational Research 175 (2), 672–687.
Masson, R., Trentini, A., Lehuede, F., Malhene, N., Peton, O., Tlahig, H., 2015. Optimization of a city logistics transportation system with mixed passengers and goods. EURO Journal on Transportation and Logistics 1–29.
Nanry, W., Barnes, J., 2000. Solving the pickup and delivery problem with time windows using reactive tabu search. Transportation Research Part B: Methodological 34 (2), 107–121.
Pisinger, D., Røpke, S., 2007. A general heuristic for vehicle routing problems. Computers & Operations Research 34 (8), 2403–2435.
Powell, W., Sheffi, Y., Nickerson, K., Butterbaugh, K., Atherton, S., 1988. Maximizing profits for north american van lines' truckload division: A new fremwork for pricing and operation. Interfaces 1 (18), 21–41.
Røpke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science 40 (4), 455–472.
Santos, G., Behrends, H., Teytelboym, A., 2010. Part II: Policy instruments for sustainable road transport. Research in Transportation Economics 28, 46–91.
Schilde, M., Doerner, KF., Hartl, RF., 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Computers & Operations Research 38 (12), 1719–1730.

Secomandi, N., Margot, F., 2009. Reoptimization approaches for the vehicle-routing problem with stochastic demands. Operations Research 57 (1), 214–230.
SmartLogisticsLab, 2016. Official webpage (accessed on 11.01.2016) URL http://www.smartlogisticslab.nl.
Trentini, A., Malhene, N., 2010. Toward a Shared Urban Transport System Ensuring Passengers & Goods Cohabitation. Trimestrale del Laboratorio Territorio Mobilita e Ambiente - TeMALab, pp. 37–44.
Verweij, B., Ahmed, S., Kleywegt, A., Nemhauser, G., Shapiro, A., 2003. The sample average approximation method applied to stochastic routing problems: a computational study. Computational Optimization and Applications 24, 289–333.