Crama, Yves; Pironet, Thierry Léon Augustin

Article

Vehicle allocation problem with uncertain transportation requests over a multi-period rolling horizon

Logistics Research

Provided in Cooperation with:
Bundesvereinigung Logistik (BVL) e.V., Bremen

Suggested Citation: Crama, Yves; Pironet, Thierry Léon Augustin (2020) : Vehicle allocation problem with uncertain transportation requests over a multi-period rolling horizon, Logistics Research, ISSN 1865-0368, Bundesvereinigung Logistik (BVL), Bremen, Vol. 12, Iss. 1, pp. 1-13, https://doi.org/10.23773/2019_1

This Version is available at:
https://hdl.handle.net/10419/297168

Logistics Research (2019) 12:1
DOI_10.23773/2019_1

Vehicle allocation problem with uncertain transportation requests over a multi-period rolling horizon

Y. Crama, T. L. A. Pironet

Received: 13 November 2017 / Accepted: 30 December 2018 / Published online: 4 February 2019

ABSTRACT

This work investigates optimization techniques for a multi-period vehicle allocation problem with uncertain transportation requests revealed sequentially over a rolling horizon. Policies derived from deterministic scenarios are compared: they are generated either by simple heuristics, or by more complex approaches, such as consensus and restricted expectation algorithms, or by network flow formulations over subtrees of scenarios. Myopic and a posteriori deterministic optimization models are used to compute bounds allowing for performance evaluation and for estimating the value of information. The economic benefit of the stochastic model is highlighted: our results show that the information about future, uncertain orders contained in the stochastic part of the horizon can be used to generate improved profits. Robustness against misspecified probability distributions is examined. Subtree formulations produce the best results, are robust and can be solved efficiently, which makes them appropriate for industrial implementations.

„This article is part of a focus collection on ‘‘Dynamics in Logistics: Dynamics in Maritime and Transport Logistics“

Yves Crama
QuantOM, HEC-Management School,
University of Liège
Tel.: +0032-43663077
E-mail: yves.crama@uliege.be
ORCID 0000-0002-7470-4743

Thierry Léon Augustin Pironet
QuantOM, HEC-Management School,
University of Liège
Tel.: +0032-43662396
E-mail: thierry.pironet@uliege.be
ORCID 0000-0003-1332-8964

KEYWORDS: transportation · vehicle allocation · pick-up and delivery · multi-period · stochastic

1 INTRODUCTION

In this paper, we investigate a Dynamic Vehicle Allocation problem (DVAP) faced by forwarding companies active in road transportation. We assume that a company owning a limited fleet of vehicles attempts to maximize its operational profit over an infinite horizon divided into equal periods (typically, days). A decision leading to a set of actions is made at every period and is based on the dispatcher's information over a restricted rolling horizon (typically, one week) as the dispatcher cannot foresee the transportation requests in the tail of the horizon. The profit stems from revenues collected when transporting full truckloads (FTL), after taking into account all costs incurred when waiting idle (during dwell time) or when moving unladen. The data revealed over time by the clients relate to their prospective orders, or transportation requests: locations of pick-up and destination cities, and a unique pick-up period for each order. Moreover, the dispatcher can rely on data regarding travel times between cities, current location and status (unladen or loaded) of each truck. This information is known with full certainty and represents the deterministic part of the problem.

The stochastic component of the problem arises from the uncertainty on the transportation requests. More precisely, for order forecasts in the remote part of the rolling horizon, the dispatcher only knows the order confirmation probability. The availability of each transportation order is either confirmed, or denied by the clients, a few periods ahead of the loading period. When a client confirms an order, the carrier may still decide to fulfill it with its own fleet or to subcontract it.

Therefore, the decision problem faced by the dispatcher in each period is either to select or to outsource confirmed orders for this period, then to allocate the selected orders to its trucks, taking into account the availability and current location of the

fleet as well as prospective and confirmed orders for future periods of the rolling horizon. Subcontracting is assumed to take place at no cost while bringing no profit (meaning that the contract is simply transferred to the subcontractor).

The main objective of our research is to provide generic, practical, yet effective algorithmic strategies to tackle this multi-period stochastic vehicle allocation problem over a rolling horizon. The problem is computationally difficult due to the large number of possible realizations of the random variables and to the combinatorial nature of the decision space. Our methodology is based on optimizing decisions for deterministic scenarios, so as to alleviate the stochastic aspect of the problem. By solving the allocation problem for a sample of scenarios, by mixing solutions, or by evaluating them through a look-ahead procedure, we aim at selecting actions which generate profit in the long run. The nature of our contributions will be described in more detail in the next section, after a short literature review.

2 LITERATURE REVIEW AND CONTRIBUTIONS

In its simplest deterministic version where the horizon is finite and where all orders are known with certainty, the Dynamic Vehicle Allocation problem has been studied for several decades, in particular in connection with the problem of repositioning empty vehicles (mostly, rail cars) in a transportation network; see, e.g., the surveys ([7], [8]). Several authors have observed that this version of the problem can be formulated as a min-cost network flow problem and hence, can be solved in polynomial time (see Section 4 hereunder).

Papers [28] and [32] have used and solved such formulations in a deterministic rolling-horizon framework which follows the generic description provided in [26]. Their main objective was to examine the influence of the length of the rolling horizon, the order density, the trip length and the fleet size on the quality of the solutions obtained.

The deterministic model can also be viewed as a special case of the generic pick-up and delivery problems (PDP) discussed in [23], with a profit maximization objective as in [20] or [24]. Our formulation only considers full truckloads, whereas the PDP literature usually allows less-than-full truckloads. Moreover, our DVAP does not include sideconstraints such as driver regulations, mandatory return to the depot, or time windows.

In its dynamic and stochastic version, the DVAP has been almost exclusively studied by Powell and his coauthors in a series of papers: [11], [16], [17], [18], [19], [27], [29]. Crainic in [7] provides a nice survey of these contributions. These papers propose several approaches, such as Approximate Dynamic Programming (ADP), for the solution of the DVAP and of its extensions to more abstract resource allocation models. Most of these approaches are based on solving a sequence of multistage (typically, two-stage) problems with recourse, where the recourse accounts for all future periods after the first one(s). As observed by [7], the resulting models are increasingly complex. A main difficulty lies clearly in the determination of appropriate recourse value functions. In the applications considered by Powell et al., both the number of clients to be served in each node of the transportation network and the number of trucks are very large. This makes it possible to rely on concepts like "the marginal value of serving a request", or "the marginal value of relocating a truck in a node". As observed in [32], the models developed in this line of research actually pay little attention to the construction of continuous truckload routes, but rather concentrate on the availability of trucks in nodes where demand is likely to be expressed. Our focus, on the other hand, is on applications where a medium-size carrier faces demand that may be scarce and geographically scattered. In such settings, tour feasibility for individual trucks becomes a crucial feature, and purely combinatorial algorithms appear to be much better suited for this type of situations. This is the approach adopted, in particular, in [28], [32], and in our work. In this framework, we pay special attention to the evaluation of the relative performance of the heuristics, and to the estimation of the value of information, as suggested for instance in the literature review by [32]. Indeed, evaluating the performance of multi-period or stochastic optimization algorithms is difficult, since the value of the optimal policy is usually not known. Therefore, researchers frequently rely on comparisons with (estimates of) lower bounds derived from myopic strategies, as in [2] or [31], or on a posteriori (optimistic) upper bounds, as in [15], [25] or [27]. It is not common to find both myopic and a posteriori bounds in the same article; an example for the deterministic multi-period setting can be found in [28]. In [4], the authors observe that, presently, there exists no standard performance analysis framework for dynamic stochastic problems (such as computing time or number of iterations for static and deterministic problems), and they stress the need to compute meaningful bounds.

With respect to the literature reviewed above, our contributions can be outlined as follows. First, in Section 4, we add a stochastic dimension to the multi-period, rolling-horizon settings of [28] and [32], so as to model the uncertainty regarding surrounding future transportation requests. In Section 5, we propose several heuristic procedures for the solution of the resulting stochastic decision problem. These heuristics are based on the solution of sub-problems associated with deterministic scenarios. They are customizations of generic strategies described, for instance, in [5], [13], or [30], and are tailored to the specific problem at hand. Our work also contributes to the evaluation of the relative performance of the heuristics: in Section 7.2,

we describe and compute various bounds, which are then used to evaluate and to compare the performance of our algorithms. The analysis of the computational results is performed in a statistical setting which allows us to validate the conclusions in a meaningful way (Section 7.3). As a by-product of these contributions, we also underline in Section 8 the managerial benefits that can be drawn by explicitly taking into account the multi-period and stochastic nature of the problem: namely, we provide illustrative numerical estimations for the expected values of the perfect information (EVPI), of the expected value solution (EEVS), and of the stochastic solution (EVSS). Finally, we analyze the robustness of our algorithms with respect to the accuracy of the estimation of the probability distributions.

3 PROBLEM STATEMENT

3.1 Complete horizon

A long-haul transportation company attempts to maximize its profit by delivering transportation requests, referred to as *loads* or *orders*, which it can assign to its own fleet or outsource to partner companies. The set of potential orders is denoted as L. Each order is transported individually by a truck (Full-Truck-Load, or FTL). The fleet is homogeneous and limited: it consists of I identical trucks.

The set of decision periods $\{1, 2, ..., WH\}$ stands for the horizon of the company. Typically each period represents a day, and the number of periods in the horizon is very large, essentially endless in practice. We assume that the complete problem over these WH periods is handled by solving a sequence of sub-problems, where each sub-problem is defined by data regarding the transportation orders over a rolling horizon of length $H+1$ (for instance, over a week). No information is available about future transportation requests in remote periods beyond the rolling horizon of length $H+1$. (Depending on the respective values of WH and H, these assumptions allow us to model the information about demand over a part of the horizon only, or over the complete horizon. But typically, we think of H as being much smaller than WH, and we do not formulate any strong hypothesis, like stationarity, about the order generating process beyond period $H+1$.) In [28], the authors refer to this setting, where the carrier relies on advance information obtained, for example, from clients or from market places, as *dispatching with look-ahead*. Hence, the periods taken into account in each sub-problem are of the form $\{t, ..., t+H\}$ where $t \in \{1, ..., WH-H\}$ stands for the current *decision period*. (The last decision period to be considered in the rolling horizon process is period $WH-H$.)

Each order $j \in L$ transported by the company must be loaded and unloaded on fixed dates, predetermined by the client: order j must be loaded at the beginning of the *pick-up period* $a_j \in \{1, 2, ..., WH\}$, and unloaded at the end of the *delivery period* $b_j \geq a_j$. (This may reflect a JIT environment, where pick-up and delivery dates must be strictly respected.)

Origins and destinations of orders are located in a restricted set C of nodes (say, cities) of a network. The pick-up and destination cities of order j are α_j and β_j, respectively. The trip duration for load j is equal to the number of periods needed to travel from α_j to β_j. The distance between each pair of cities is expressed as an integer number of periods. So, the trip duration d_j is equal to the distance between α_j and β_j, also denoted as $d_j = d(\alpha_j, \beta_j)$. We assume that $a_j + d_j - 1 = b_j$, reflecting the fact that the load is delivered without delay.

The uncertainty of the forecast regarding the availability of a specific transportation order j is modeled by a Bernoulli distribution:

$$P(q_j = x) = \begin{cases} p_j & if\ x = 1 \\ 1 - p_j & if\ x = 0 \end{cases} \quad (1)$$

where q_j is a random variable which takes value 1 if order j is released (i.e., if the transportation order is confirmed by the client), and p_j is a parameter in [0, 1]. The choice made by the company either to transport with its own trucks or to outsource an order in period t can be influenced by information pertaining to the stochastic part of the horizon, which might not be confirmed a posteriori. We now turn to a more accurate description of this rolling horizon setting and of the timing of the information.

3.2 Single rolling horizon

Consider a rolling horizon $\{t, ..., t+H\}$. At decision period t, truck i is located at node $\gamma_i(t)$ of the network. Some trucks are unladen (available to be loaded) and some are already loaded (transporting an order). Loaded trucks are unavailable for a new order allocation, and will only become available in a subsequent period after having delivered their current order. The orders for which $a_j = t$ are available for loading. If the decision is made to load order j, then the order is allocated to an unladen truck i which must be present and available at location α_j at time t, so if $\gamma_i(t) = \alpha_j$.

We denote by RH the number of upcoming periods that contain deterministic information about the availability of orders. So, the information included in periods $\{t, ..., t+RH\}$ is fully revealed. (In other words, the confirmation or cancellation of an order j with release date $a_j = t+RH$ is revealed in period t.) All transportation orders j such that $a_j \in \{t+RH+1, ..., t+H\}$ are *projections*, or *forecasts*, meaning that the dispatcher can anticipate their features (release date, delivery date, etc.), but is not sure whether the orders will be confirmed by the clients, or not. All decisions regarding orders (i.e. to transport or to outsource) with a_j in $\{t+1, ..., t+H\}$ might be revised

until a_j becomes the decision period (see [28] and [32] for similar settings). Also, waiting and unladen moves decisions are reevaluated at each successive decision period.

In summary, at time t, the dispatcher must decide for each unladen truck i positioned at node $\gamma_i(t)$:

(a) either to load on truck i any order j such that $a_j = t$ and $\gamma_i(t) = \alpha_j$; then, truck i starts moving towards node β_j, where it will arrive at time b_j; truck i will be available again for loading at the start of period $b_j + 1$;
(b) or to get truck i moving (unladen) towards another node $c \in C$;
(c) or to keep truck i waiting at its current location: $\gamma_i(t) = \gamma_i(t+1)$ (this can also be viewed as a special case of the previous decision).

The total profit to be maximized results from profits earned when delivering orders and from costs generated by unladen trips or by waiting on site. Related parameters are:

– g_j: profit per period for transporting order j (equal for all trucks),
– e: cost per period of an unladen trip (equal for all trucks),
– f: waiting cost per period (equal for all trucks in all locations).

We assume that $g_j > 0$ and $e > f > 0$.

As there is a selection process, those orders with $a_j = t$ that are not allocated to a truck of the company in period t are subcontracted, and bring neither revenues, nor penalties. Since the objective of our model is profit maximization, this simply means that the cost of subcontracting an order is the foregone revenue. A similar assumption is made, for instance, in [31] and [32]. As in [20] or [24], one can assume that another trailer company, working as a subcontractor or in the framework of a broader collaborative agreement, can be activated for any single order during the decision period t at no cost for the carrier. This assumption implies that all plans are feasible and that no penalty costs are incurred due to past decisions. The transportation industry frequently features large excess capacity (small average vehicle loads and empty return trips), as confirmed in [10]. The authors of [32] also mention that "last minute call for transportation services is very common in the industry". Electronic exchange platforms actually facilitate this type of calls to the spot market. When the assumption regarding the outsourcing of orders is considered to be too flexible, the model could be easily modified either to assume that decisions are frozen over a subhorizon of the deterministic horizon $\{t, ..., t + RH\}$ including more than one period, or to include a penalty term in the objective function when an order is outsourced on short notice, but we did not implement these modifications in our models.

Note that, in a myopic setting (say, with $H = 0$), no available order would ever be discarded when the available fleet is sufficient, since the alternative option would consist in moving unladen or in waiting on site, and since revenues are preferred to costs. In the multi-period model, however, this conclusion does not necessary hold: it may be more profitable to wait or to move unladen in order to pick up a future order, rather than to load an available order. In such cases, comparisons have to be made between various combinations of loading actions, waiting times and unladen trips. The combinatorial aspect of the decision problem comes from this blend of feasible actions.

Let us conclude this subsection with a few comments regarding the parameters and assumptions of the model for a single rolling horizon.

Remark 1. Sometimes, the decision for a truck i in the current decision period t may be to move from its current location $\gamma_i(t)$ towards a different city α_j, with a view to loading order j in a future period $a_j = t + k$. This might imply for the truck to move unladen, but also to spend some time waiting at destination or along the way, in case the travel time to α_j is shorter than k. In such a case, we assume that the truck always prefers to wait at $\gamma_i(t)$ rather than to move, so as to avoid potentially useless unladen trips (e.g., if the initial decision is modified after period t, the truck might have to go back to its initial position). The reverse option would lead to a repositioning strategy which is not in the scope of this research.

Moreover, still in order to avoid useless moves, we define a parameter $D \leq RH$ such that unladen trips of length larger than D are not allowed. This avoids useless trips if a projected order j does not become available. Both restrictions on useless trips might hurt the expected profit, but they are derived from practice, as drivers do not appreciate to drive to a place where an expected transportation order may eventually be canceled (see [19]).

Remark 2. Each unladen trip between two cities (departure city $\gamma_i(t)$, arrival city α_j) consists of a sequence of cities along a path. This path is taken to be the shortest one (ties are arbitrarily broken). In fact, in a rolling horizon framework, what matters is only the location of the unladen truck on this path after one period, as this location is updated at the end of the decision period and a new decision is made in the next period for the truck. Without loss of generality, by adding enough (fictitious) cities or locations in the model, we can assume that the truck is always located in a city at the end of each period.

3.3 Sequence of rolling horizons

At each decision period t, the rolling procedure includes information about order j if the pick-up period a_j is included in the rolling horizon $\{t, ..., t+H\}$. This implies that the subset of orders under consideration keeps changing from decision period to decision period. This subset is labeled J_0 (the index t is omitted for short): $J_0 = \{j \in L : t \leq a_j \leq t + H\}$. The availability of order

Table 1: Model parameters

Parameters	Explanations
WH	Number of periods in the complete horizon
H	Number of upcoming periods in the rolling horizon
RH	Number of upcoming periods in the revealed (deterministic) rolling horizon
t	Current decision period
I	Fleet size
I_0	Set of unladen trucks at the start of period t
I_1	Set of loaded trucks at the start of period t
$\gamma_i(t)$	Location of truck i at the start of period t
J_0	Set of confirmed and projected orders with pick-up period in the rolling horizon
J_0^s	Set of orders confirmed in scenario s with pick-up period in the rolling horizon
J_1	Set of orders initially loaded on trucks at the start of period t
a_j	Pick-up period of order j
b_j	Delivery period of order j
α_j	Pick-up city of order j
β_j	Delivery city of order j
$d(x,y)$	Distance from city x to city y
d_j	Trip duration for order j: $d_j = d(\alpha_j,\beta_j) = b_j - a_j + 1$
p_j	Probability of confirmation of order j
g_j	Profit per period for transporting order j
e	Cost per period when moving empty
f	Cost per period when staying idle
v_j	Truck allocated to order j ($v_j = 0$ if j is not loaded)
c^1	First city on a path from $\gamma_i(t)$ to α_j
D	Maximum number of periods for an empty move

j (i.e., the value of q_j) is fixed as soon as a_j enters the "revealed horizon", that is, in the period $t = a_j - RH$. In subsequent periods, q_j remains fixed at the same value.

Previous allocations of trucks must be taken into account at every time. A truck-order allocation is represented by a parameter v_j: if truck i is allocated to order j, then $v_j = i$.

At the beginning of decision period t, the subset of orders which have been previously loaded and which are currently being transported is: $J_1 = \{ j \in L : a_j < t, b_j \geq t, v_j \neq 0 \}$. The subset of loaded trucks at time t is denoted by I_1.

When $v_j = i$, the truck i is allocated to order j temporarily, from loading to unloading. After unloading order j, the allocation parameter is reset (v_j=0), truck i is "free for loading" and belongs to the subset of unladen trucks $I_0 \subseteq \{1, .., I\}$. Obviously, $|I_0| + |I_1| = I$ and $|J_1| = |I_1|$.

At each period t, the state of the system evolves as follows:

1. New orders j such that $a_j = t + H$ enter the rolling horizon and consequently enter the set J_0.
2. Orders j such that $a_j = t + RH$ enter the fully revealed horizon, meaning that the value of q_j is fixed permanently; if $q_j = 0$, then order j can be removed from J_0.
3. Orders j such that $a_j < t$ can be removed from J_0.
4. For each truck i, an action is taken, which can be:
   (a) if truck $i \in I_0$, either to load some order $j \in J_0$; then, $v_j := i$, order j is placed in J_1, and truck i is placed in I_1;
   (b) or if truck $i \in I_0$, to move unladen towards α_j, the loading city of an order $j \in J_0$; then, at the end of period t, truck i is repositioned in the first city c^1 on the path from $\gamma_i(t)$ to α_j;
   (c) or if truck $i \in I_0$, to wait in city $\gamma_i(t)$;
   (d) or if truck $i \in I_1$, to carry on if $b_j > t$ or to unload order $j \in J_1$ if $b_j = t$ as a consequence of a previous decision, where $v_j = i$; recall that the destination β_j is reached at the end of the period and the truck is unloaded in the same period; so, before the start of the next period, the allocation parameter v_j is reset to 0, order j is removed from J_1, and truck i is placed in I_0.

The expected profit μ_π per period for a decision policy π over the entire horizon $\{1, ..., WH\}$ (where we think of WH as being much larger than H, or $WH \to \infty$) is defined as

$$\mu_\pi = \frac{1}{WH - H} E\Big[\sum_{t=1}^{WH-H} C(S_t, A^\pi(S_t)) \Big], \tag{2}$$

where S_t denotes the state of the system at time t (defined by the collection of orders available for shipment at time t, deterministic and stochastic information over the horizon $\{t+1, ..., t+H\}$, the

truck locations and their status – loaded or not), $A^{\pi}(S_t)$ denotes the actions taken in state S_t according to policy π, and $C(S_t, A^{\pi}(S_t))$ is the associated profit collected in period t (profits minus costs incurred at t). The objective is to maximize μ_{π} over all feasible policies π.

4 FORMULATION

In this section, we present integer programming formulations of various deterministic versions of our vehicle allocation problem. These formulations will prove useful when solving the general stochastic version of the DVAP. The main parameters of the models are displayed in Table 1.

4.1 Scenario-based deterministic models

The information regarding the availability of a transportation order $j \in J_0$ is deterministic when $a_j \in \{t, ..., t+RH\}$; it is stochastic and modeled by the Bernoulli variable q_j when $a_j \in \{t+RH+1, ..., t+H\}$. A *scenario* s is defined by fixing a deterministic value $q^s_j \in \{0, 1\}$ of the random variable q_j for each order j with $a_j \in \{t+RH+1, ..., t+H\}$. So, the probability W_s of occurrence of scenario s is

$$W_s = \prod_{a_j \in \{t+RH+1, ..., t+H\}} \left(p_j q^s_j + (1-p_j)(1-q^s_j)\right). \quad (3)$$

In each scenario s, the set of orders under consideration, say J^s_0, contains order $j \in J_0$ if and only if $q^s_j = 1$. In particular, each set J^s_0 includes the same set of orders in the deterministic part $\{t, ..., t+RH\}$ of the horizon. The set of previously loaded orders remains fixed and equal to J_1 for all scenarios. Each scenario yields a corresponding deterministic model.

A deterministic equivalent representation of DVAP could be obtained by a complete scenario tree including all possible scenarios that might arise in the rolling horizon, as explained in Section 4.4. Since the number of scenarios is huge, we propose and test hereunder several reduced, i.e., approximate, models solved to optimality. This leads to heuristic algorithms based on single or multiple scenarios, which may either be viewed as independent of each other or as forming a subtree representation of the problem.

4.2 Formulation for a mono-scenario model

In this section, we provide a formulation for a mono-scenario model, or equivalently, for a deterministic model over the horizon $\{t, ..., t+H\}$ (see, e.g., [28] for a similar formulation, which also includes costs for late delivery and subcontracting). The representation is based on a time-space graph associating locations and periods with trucks and orders.

For a set of trucks $I_0 \cup I_1$ and a subset of orders $J^s_0 \cup J_1$ in the deterministic scenario, the time-space directed graph $G = (V, E)$ is defined as follows.

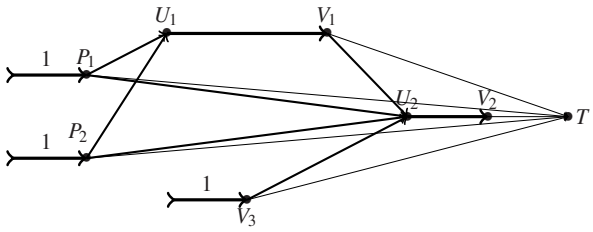

Fig.1: Graph G = (V, E):3 trucks (1 loaded) and 2 potential orders

Vertices: The set V contains a vertex P_i for each unladen truck $i \in I_0$, a vertex U_j for each available order $j \in J^s_0$, a vertex V_j for each order $j \in J^s_0 \cup J_1$, and a vertex T. We can think of the vertices in the following terms:

- P_i is associated with (the initial location of) unladen truck $i \in I_0$ at time t, $P_i = (\gamma_i(t), t)$,
- U_j is associated with the pick-up city and period of order $j \in J^s_0$, $U_j = (\alpha_j, a_j)$,
- V_j is associated with the delivery city and period of order $j \in J^s_0 \cup J_1$, $V_j = (\beta_j, b_j)$,
- T is a sink vertex associated with the end of the rolling horizon, $T = t+H$.

Note that the vertices in V should not be confused with cities: they should rather be viewed as abstract entities associated with combinations of trucks, orders, cities and periods. In the network flow model, $\{P_i : i \in I_0\}$ and $\{V_j : j \in J_1\}$ act as supply vertices. The demand or total flow at T is $I = |I_0| + |I_1|$.

Arcs: Before we define the arc set E, let us introduce two families of binary parameters that describe the feasibility of certain truck moves, as in [28]. They are respectively denoted by TL (for Truck to Load) and LL (for Load to Load). For $i \in I_0$ and $j \in J^s_0$, we set $TL(i, j) = 1$ if city α_j can be reached from the current location of truck i, namely $\gamma_i(t)$, before the loading period a_j. For $j \in J^s_0 \cup J_1$, $k \in J^s_0$, $j \neq k$, we set $LL(j, k) = 1$ if city α_k can be reached (by any truck) before period a_k after having unloaded order j at city β_j. More precisely,

$$\begin{aligned} TL(i,j) &= 1 \text{ if } d(\gamma_i(t), \alpha_j) \leq \min(a_j - t, D) \\ &= 0 \text{ otherwise,} \end{aligned} \quad (4)$$

$$\begin{aligned} LL(j,k) &= 1 \text{ if } d(\beta_j, \alpha_k) \leq \min(a_k - b_j - 1, D) \\ &= 0 \text{ otherwise.} \end{aligned} \quad (5)$$

These parameters provide information on feasible connections and can be used to reduce the arc set, and hence, the decision space. Now, the arcs in E can be divided into five categories.

- Pick-up arcs: (P_i, U_j) for $i \in I_0$ and $j \in J^s_0$ such that $TL(i, j) = 1$.

- Connection arcs: (V_j, U_k) for $j \in J_0^s \cup J_1$ and $k \in J_0^s$, $j \neq k$, such that $LL(j,k) = 1$.
- Transportation arcs: (U_j, V_j) for $j \in J_0^s$.
- End of horizon arcs: (V_j, T) for $j \in J_0^s \cup J_1$.
- Standstill arcs: (P_i, T) for $i \in I_0$.

Figure 1 provides an illustration of the structure of the graph $G = (V, E)$.

Flow variables: Each arc in E is associated with a single decision variable which represents the decision for a truck either to travel from a city to another one (unladen or loaded), or to wait. All decision variables are binary, reflecting the fact that each order is transported individually (FTL) or that the variables translate go-no go decision. They are named according to the corresponding category of arcs: PU_{ij}, UV_j, VU_{jk}, VT_j, and PT_i.

- PU_{ij} represents the decision for truck $i \in I_0$ to move from its initial location to α_j so as to pick-up order $j \in J_0^s$ (this might mean to load order j in the present truck location if $\gamma_i(t) = \alpha_j$, or to move unladen in order to pick up order j in city α_j if $\gamma_i(t) \neq \alpha_j$ and $a_j > t$; in both cases, the truck may have to wait at $\gamma_i(t)$ for a few periods if $t + d(\gamma_i(t), \alpha_j) < a_j$) (cf. Remark 1 in Section 3.2);
- VU_{jk} represents the decision for a truck to travel from the destination city of order $j \in J_0^s \cup J_1$ to the pick-up city of order $k \in J_0^s$;
- UV_j represents the decision to transport order $j \in J_0^s$;
- VT_j represents the decision for a truck to wait at β_j until the end of the horizon after delivering $j \in J_0^s \cup J_1$;
- PT_i represents the decision for a truck $i \in I_0$ to stay idle in city $\gamma_i(t)$ until the end of the horizon.

Before we turn to the definition of the cost coefficients in the profit function (Equation (6)), let us first define a few additional parameters.

Waiting times parameters: Since a decision for a truck might be to wait in its present location for at least one period, some nonnegative parameters related to waiting times can be introduced, based on truck and order information: see Table 2.

Arc costs values: The cost of each arc is denoted according to the corresponding categories: $C_{PU_{ij}}$, $C_{VU_{jk}}$, C_{UV_j}, C_{VT_j}, and C_{PT_i}. At the initial time t, some trucks $i \in I_1$ are busy transporting an order. Such loaded trucks continue their trip during the current decision period and we assume that their cost has already been taken into account. Thus, we solely include in Equation (6) the costs and profits generated by the decisions made at the current period t.

Let us now turn to a definition of the cost associated with each arc as follows:

- (Pick-up arcs) $C_{PU_{ij}}$, the cost of arc (P_i, U_j), is the cost of moving unladen and/or waiting before picking up order j, starting from the initial location of truck i:
  $C_{PU_{ij}} = -(d(\gamma_i(t), \alpha_j) * e + f * w_{ij}^0)$ for $i \in I_0$ and for $j \in J_0^s$, if $TL(i,j) = 1$.
- (Connection arcs) $C_{VU_{jk}}$, the cost of arc (V_j, U_k), is the cost of moving unladen and/or waiting before picking up order k, starting from location β_j:
  $C_{VU_{jk}} = -(d(\beta_j, \alpha_k) * e + f * w_{jk})$ for $j \in J_0^s \cup J_1$, $k \in J_0^s$, $j \neq k$, if $LL(j,k) = 1$.
- (Transportation arcs) C_{VU_j}, the cost of arc (U_j, V_j), is the profit generated by transporting order j from α_j to β_j:
  $C_{UV_j} = +d(\alpha_j, \beta_j) * g_j$ for $j \in J_0$.
- (End of horizon arcs) C_{VT_j}, the cost of arc (V_j, T), is the cost of waiting at β_j until the end of the horizon after delivering order j:
  $C_{VT_j} = -f * w_j^H$ for $j \in J_0 \cup J_1$.
- (Standstill arcs) C_{PT_i}, the cost of arc (P_i, T), is the cost of remaining idle at $\gamma_i(t)$ throughout the horizon:
  $C_{PT_i} = -f * (H + 1)$ for $i \in I_0$.

At this point, we are ready to present a network flow formulation for the mono-scenario vehicle allocation problem. Note that, since all variables are associated with arcs in E, we implicitly assume in the formulation that $PU_{ij} = 0$ when $TL(i,j) = 0$, and that $VU_{jk} = 0$ when $LL(j,k) = 0$. In fact, removing in this way all arcs and decision variables associated with infeasible links allows us to switch from a "multi-period" to a "timeless" model, as all temporal constraints are automatically satisfied (see [9]).

Objective function:

$$\max Z = \sum_{j \in J_0^s \cup J_1} \left(C_{VT_j} * VT_j + \sum_{k \in J_0^s,\, k \neq j} (C_{VU_{jk}} * VU_{jk}) \right) + \sum_{j \in J_0^s} \left(C_{UV_j} * UV_j + \sum_{i \in I_0} (C_{PU_{ij}} * PU_{ij}) \right) + \sum_{i \in I_0} C_{PT_i} * PT_i. \tag{6}$$

The maximization of Z is subject to the following constraints.

Flow conservation constraints

$$\text{Node } P_i: \sum_{j \in J_0} PU_{ij} + PT_i = 1 \text{ for } i \in I_0, \tag{7}$$

$$\text{Node } U_j: \sum_{i \in I_0} PU_{ij} + \sum_{k \in J_0^s \cup J_1,\, k \neq j} VU_{kj} = UV_j \text{ for } j \in J_0^s, \tag{8}$$

$$\text{Node } V_j: UV_j = VT_j + \sum_{k \in J_0^s,\, k \neq j} VU_{jk} \text{ for } j \in J_0^s, \tag{9}$$

$$\text{Node } V_j: VT_j + \sum_{k \in J_0^s} VU_{jk} = 1 \text{ for } j \in J_1. \tag{10}$$

Table 2: Waiting time parameters

	Explanations	Formula
w^0_{ij}	Waiting time for truck $i \in I_0$ to reach order j at period a_j	$w^0_{ij} = a_j - d(\gamma_i(t), \alpha_j) - t$ if $TL(i,j) = 1$ (undefined otherwise)
w_{jk}	Waiting time between unloading order j and loading order k	$w_{jk} = a_k - b_j - d(\beta_j, \alpha_k) - 1$ if $LL(j,k) = 1$ (undefined otherwise)
w^H_j	Waiting time after delivering order j until the end of the horizon	$w^H_j = max(0,\ t + H - b_j)$

Required flow constraint

$$\text{Node } T: \sum_{j \in J^s_0 \cup J_1} VT_j + \sum_{i \in I_0} PT_i = I. \tag{11}$$

Constraints (7) express that each unladen truck $i \in I_0$ either moves to pick up an order or stays idle. Constraints (8) ensure that if an order j is selected to be transported by the company, it must be reached by a truck which was initially idle or which has unloaded another order k. Constraints (9) impose that if order j is transported by the company, then, after unloading j, the truck either moves to pick up another order k or stops. Moreover, if order j is outsourced, then none of these actions can take place. Constraints (10) have the same interpretation as constraints (9) for those orders j which were initially loaded (i.e., $j \in J_1$). Finally, constraint (11) imposes that I trucks reach the sink node T, i.e., the total flow is equal to the number of trucks originally loaded $|I_1|$ or unladen $|I_0|$.

Since all decision variables are binary, each arc necessarily carries a 0-1 flow and there is no need for additional capacity constraints in the formulation (6)–(11).

4.3 Complexity for a single scenario formulation

The network flow formulation (6)–(11) has the integrality property and hence, is polynomially solvable (see [1]) (in our experiments, we simply rely on a commercial LP solver to obtain an optimal integer solution in a few seconds). It may be worth noting that polynomial solvability is due to some implicit restrictive assumptions of our model. Namely, the ending point at the end of the horizon is not imposed to any truck; no intermediate city has to be mandatorily visited (e.g., for refueling); no specific order has to be mandatorily transported; no waiting periods are imposed to comply with driving regulations; and so forth. Adding such constraints could make the problem theoretically harder to solve (NP-hard), as mentioned in [12].

It is also interesting to observe that the network flow formulation does not involve any time index nor city labels. Therefore, the size of the model is determined by the numbers of trucks I and orders J^s_0 taken into account in the rolling horizon, and does not directly depend on the number of periods and cities. In particular, the difficulty of the formulation is not affected by the discretization of time into small or large periods, although different discretization steps may lead to different time-space graphs, to different information updates, and hence, to different policies over the planning horizon.

4.4 Formulation for a deterministic subtree model

From the previous single-scenario formulation, it is easy to develop a formulation associated with a finite set, or subtree of ST scenarios over the horizon $\{t, ..., t+H\}$.

To generate a subtree formulation, the deterministic formulation of Section 4.2 is replicated ST times with scenarios indexed by $s \in \{1, ..., ST\}$.

In order to simplify the presentation, we momentarily assume that the set of orders under consideration, i.e., J_0, includes all orders in the rolling horizon. So, the graph and the variables introduced in Section 4.2 are replicated ST times except for the sink node T which remains unique.

Subtree variables: for $s = 1, ..., ST$,

- $PU^s_{ij} \in \{0,1\}$ for $i \in I_0$ and $j \in J_0$, if $TL(i,j) = 1$;
- $VU^s_{jk} \in \{0,1\}$ for $j \in J_0 \cup J_1$ and $k \in J_0$, $j \neq k$, if $LL(j,k) = 1$;
- $UV^s_j \in \{0,1\}$ for $j \in J_0$;
- $VT^s_j \in \{0,1\}$ for $j \in J_0 \cup J_1$;
- $PT^s_i \in \{0,1\}$ for $i \in I_0$.

The subtree formulation is now (compare with the single-scenario formulation (6)–(11)):

Objective function:

$$\max Z = \sum_{s=1,...,ST} \left[\sum_{j \in J_0 \cup J_1} \left(C_{VT_j} * VT^s_j + \sum_{k \in J_0;\ k \neq j} (C_{VU_{jk}} * VU^s_{jk}) \right) + \sum_{j \in J_0} \left(C_{UV_j} * UV^s_j + \sum_{i \in I_0} (C_{PU_{ij}} * PU^s_{ij}) \right) + \sum_{i \in I_0} C_{PT_i} * PT^s_i \right] \tag{12}$$

subject to the following constraints

Flow conservation constraints

$$\text{Node } P_i^s: \sum_{j\in J_0} PU_{ij}^s + PT_i^s = 1$$
$$\text{for } s = 1,\ldots,ST \text{ and for } i \in I_0, \quad (13)$$

$$\text{Node } U_j^s: \sum_{i\in I_0} PU_{ij}^s + \sum_{k\in J_0\cup J_1;\ k\neq j} VU_{kj}^s = UV_j^s$$
$$\text{for } s = 1,\ldots,ST \text{ and for } j \in J_0, \quad (14)$$

$$\text{Node } V_j^s: UV_j^s = VT_j^s + \sum_{k\in J_0;\ k\neq j} VU_{jk}^s$$
$$\text{for } s = 1,\ldots,ST \text{ and for } j \in J_0, \quad (15)$$

$$\text{Node } V_j^s: VT_j^s + \sum_{k\in J_0} VU_{jk}^s = 1$$
$$\text{for } s = 1,\ldots,ST \text{ and for } j \in J_1. \quad (16)$$

Required flow constraints

$$\text{Node } T: \sum_{j\in J_0\cup J_1} VT_j^s + \sum_{i\in I_0} PT_i^s = I \quad \text{for } s = 1,\ldots,ST. \quad (17)$$

In view of the value assumed by the random variables in each scenario, some decision variables can be removed from the model, namely: for all $s = 1, \ldots, ST$ and for all $j \in J_0$ such that $q^s = 0$,

$$PU_{ij}^s = 0 \quad \text{for } i \in I_0, \quad (18)$$
$$UV_j^s = 0, \quad (19)$$
$$VU_{jk}^s = 0 \quad \text{for } k \in J_0 \text{ and } k \neq j, \quad (20)$$
$$VU_{kj}^s = 0 \quad \text{for } k \in J_0 \cup J_1 \text{ and } k \neq j, \quad (21)$$
$$VT_j^s = 0. \quad (22)$$

As a unique set of actions is to be performed, non-anticipativity constraints must be added to ensure the consistency of the solutions over the different scenarios. Since we implement our model in a rolling horizon framework, we have chosen to express only those non-anticipativity constraints that ensure the consistency of actions taken at the current decision period t. Indeed, over the next decision step in period $t + 1$, new actions will be taken and all previous decisions will be reoptimized. Hence, fixing common actions within scenarios for periods $\{t+1, \ldots, t+H\}$ is not mandatory to ensure consistency at period t, and omitting such constraints simplifies the model. (Note that under these simplifications, even a complete tree containing all possible scenarios would only give rise to an approximate model of the multi-period stochastic problem, rather than an exact deterministic equivalent model.) Let us now explain this point in more detail.

Now, for any scenario s, in the current period t, each truck $i \in I_1$ is on the move, and will just keep moving on until (at least) the next period. Each truck $i \in I_0$ is located at $\gamma_i(t)$ (associated with node P_i^s), and all relevant, or "interesting" decisions at period t pertain to such trucks. The decisions can be of one of four types: for truck $i \in I_0$ (recall that I_0 contains all empty trucks as well as trucks unloaded at the end of the previous period), one can decide

(a) either to load an item $j \in J_0$ such that $\alpha_j = \gamma_i(t)$ and to start moving toward β_j, in which case $PU_{ij}^s = 1$ (and $UV_j^s = 1$ follows as a consequence of Equation (14));
(b) or to start moving unladen toward α_j such that $t + d(\gamma_i(t), \alpha_j) = a_j$ and $TL(i, j) = 1$, in which case $PU_{ij}^s = 1$;
(c) or to wait for at least one period before moving unladen toward α_j such that $TL(i, j) = 1$ and $t + d(\gamma_i(t), \alpha_j) > a_j$, in which case $PU_{ij}^s = 1$ (cf. Remark 1 in Section 3.2);
(d) or to wait at location $\gamma_i(t)$ until the end of the horizon, in which case $PT_i^s = 1$.

All other variables (VU_{jk}^s, UV_j^s, VT_j^s) are associated with actions to be implemented in subsequent periods.

So, in order to ensure consistent decisions, we only have to force the replicated variables PU_{ij}^s to be equal in all scenarios when the travel time is exactly equal to $a_j - t$, as in cases (a) or (b) above; in all other cases, truck $i \in I_0$ has to wait anyway and there is no need to distinguish among those different cases. In summary, the subset of non-anticipativity constraints that we include in the model (and which bear on the decisions taken in period t) are expressed by conditions (23) hereunder.

Non-anticipativity constraints for an unladen truck i in period t:

$$PU_{ij}^1 = PU_{ij}^s \text{ for } i \in I_0,\ j \in J_0, \text{ for } s = 2,\ldots,ST$$
$$\text{and if } t + d(\gamma_i(t), \alpha_j) = a_j \text{ and } a_j \leq t + D. \quad (23)$$

After solving the optimization model (12)–(23) over the rolling horizon, actions for the current period t can be extracted to generate the value of the policy for this decision period. Given that loading operations, unladen movements and waiting decisions are consistent in period t, analyzing one single scenario's variables (e.g., for $s = 1$) is enough to recover the profit of the policy during this decision period. Practically, analyzing for each unloaded truck i whether PU_{ij}^1, or $PT_i^1 = 1$ is enough to deduce the cost or profit due to truck i during period t. Consequently, at each decision step, the policy evaluation is similar to what is done for the single scenario representation.

4.5 Complexity for a subtree of scenarios

We mentioned in Section 4.3 that the DVAP is polynomially solvable for a single scenario. On the other hand, we do not know the exact complexity of the

subtree formulation. Because of the non-anticipativity constraints (23), this formulation can be viewed as an equal flow problem, and such problems are known to be generally NP-hard, except in specific cases (see [14]). In practice, however, we observed that we always obtain an optimal integer solution when solving the linear relaxation of the formulation (12)–(23), even for large instances (see Table 10). For the instances that we handled, the computing time remained within a few seconds (see Section 8.1), which is practically convenient. This is an important feature of our model since for other stochastic transportation problems, subtree formulations including integer variables may turn out to be computationally intractable (see, e.g., [3]). Therefore, we did not develop a specific algorithm to solve the subtree formulation.

5 ALGORITHMS

In a rolling horizon framework, the sequence of decisions implemented in successive periods results from solving a sequence of optimization models using deterministic and stochastic information over a sequence of restricted horizons of length $H+1$. In this section, we describe several heuristic algorithms, based on the exact solution of approximate models, that can be used to compute (hopefully good) policies.

5.1 Mono-scenario algorithms

As suggested in Section 4.2, any scenario over $\{t, ..., t+H\}$ yields a computationally efficient heuristic. We next detail some specific choices of scenarios.

5.1.1 Optimistic scenario algorithm: Opt

In this scenario, all orders $j \in J_0$ are supposed to become available for transportation, i.e., q_j is considered equal to 1 for all $a_j \in \{t+RH+1, ..., t+H\}$. The policy issued from this scenario is called *Optimistic*.

5.1.2 Modal value scenario algorithm: Mod

This scenario generates a policy based on the modal value of the probability distribution: if $p_j \geq 0.5$, the scenario includes j ($q_j = 1$), otherwise, order j is not included in the scenario ($q_j = 0$). This scenario, like the optimistic one, may lead to extreme situations, i.e., discarding or keeping orders even if their probability of occurrence is close to 50%. Moreover, the expected profit of each order is not taken into account, but only its availability probability.

5.1.3 Expected Value Scenario algorithm: EG

To avoid the drawbacks of the previous scenarios, a policy can be derived from a heuristic based on a modification of the optimistic scenario. This "pseudo-scenario" includes all orders $j \in J_0$, as in *Opt*, but the profit per period is set equal to its expectation $g'_j = g_j * p_j$ instead of g_j for each order j with $a_j \in \{t + RH + 1, ..., t + H\}$. Strictly speaking, the resulting deterministic instance is not really derived from a scenario in the sense of Section 4.1 (since the profits have been modified). Nevertheless, for the sake of simplicity, we refer to it as the *expected value scenario*. The value of the optimal policy for this scenario is called the "Expectation of the Expected Value Solution", or EEVS (see [5]).

5.2 Multiple-scenario algorithms

To find a single scenario leading to a good policy requires some luck. Another strategy might be to consider K random independent scenarios, called *calibration scenarios*, and to select or to generate a policy derived from the K corresponding solutions.

Two particular optimization methods based on multiple scenarios are described hereunder: a *Consensus algorithm Cs* and a *Restricted Expectation algorithm RE**. These methods rely on generic strategies for algorithmic design which, however, must be significantly tailored to the problem at hand (see [30] for a description of the generic principles, and [3] for an application). In each case, we assume that K calibration scenarios have been generated by Monte Carlo simulations based on the probability distributions of the availability variables q_j, for all orders j with $a_j \in \{t+RH+1, .., t+H\}$.

5.2.1 Consensus algorithm: Cs

The deterministic models (6)-(11) associated with K calibration scenarios are solved independently, and the actions (load, move unladen, wait) for the current period t are recorded as θ_k for $k = 1, ..., K$. The aim of the consensus algorithm *Cs* is to generate a new "compromise" solution based on the most frequent decisions taken among the solutions θ_k. At first sight, a common decision means that corresponding variable values have to be the same in different solutions. Yet, identical decisions can also be derived from various variables in models associated with different scenarios. For instance, in a solution $\theta_{k'}$, the decision for a truck to wait in city c can be issued from a variable representing a complete standstill decision over the rolling horizon while in another solution θ_k, $k \neq k'$, the decision for this truck to wait might be issued from the plan to load in city c in the next period. Conversely, seemingly different, but globally equivalent decisions can be taken within two solutions $\theta_{k'}$ and θ_k. Consider for instance, two idle trucks located at the same time in the same city. In solution $\theta_{k'}$, truck A waits, truck B loads, and conversely in solution θ_k. These decisions are equivalent, even if the replicated variables are not indexed in the same way. So, there is a need for an aggregation phase of variables representing identical or equivalent decisions. These equivalent decisions can then be used to build a consensus solution. This gives rise to two distinct phases in our consensus algorithm.

First, in **the aggregation phase**, we generate several counters for each city $c \in C$ for each kind of decision, namely, loading, moving unladen, or waiting. At the current period t, the counters are:

- $CL_{(c,j)}$, number of solutions $\theta_1, ..., \theta_K$ such that order $j \in J_0$ is loaded in city $c \in C$ in period t;
- $CE_{(c,c^1)}$, total number of trucks planned to move unladen from city c to city c^1 in period t over all solutions $\theta_1, ..., \theta_K$;
- CW_c, total number of trucks planned to wait in city $c \in C$ in period t over all solutions $\theta_1, ..., \theta_K$.

Next, all loading operations and all unladen moves are aggregated per city c, so that the most frequent decisions among loading, moving unladen or waiting can be ranked per city. The resulting counters are divided by K and rounded to the nearest integer to get the number of decisions of each type per city regardless of parameter K. These integer values provide the number of similar decisions to be allocated to trucks located in each city, namely:

- $NL_c = [\frac{1}{K} \sum_{j \in J_0} CL_{(c,j)}]$ number of trucks that should load in city c,
- $NE_c = [\frac{1}{K} \sum_{c^1 \in C} CE_{(c,c^1)}]$ number of trucks that should move unladen out of city c,
- $NW_c = [\frac{1}{K} CW_{(c)}]$ number of trucks that should wait in city c.

This ends the aggregation phase of decisions.

The allocation phase consists of two parts. First, for each city c and for each idle truck currently located in c, we allocate an operation according to an iterative procedure considering the counters NL_c, NE_c or NW_c in non-increasing order. For instance, if $NL_c \geq NE_c$ and $NL_c \geq NW_c$, then the first truck should load an order. Each time an action is allocated to a truck, the value of the current indicator NL_c, NE_c or NW_c is reduced by one. Therefore, for the next truck another kind of decision might be allocated. This first phase of the allocation procedure is repeated iteratively for each truck in each city.

The second phase of the allocation procedure aims at defining which order j should be loaded when a loading decision has been allocated to a truck and which destination city c^1 should be selected when an unladen move has been allocated. To select the order j to be loaded, loading counters from city c are ranked by non-increasing values of $CL_{(c,j)}$ and each order is allocated once subsequently. Similarly, unladen moves are specified according to non-increasing values of $CE_{(c,c^1)}$. The current largest counter value is reduced by one after each allocation and counters are resorted accordingly.

In case of ties among NL_c or NE_c or NW_c, preference is given first to loading, then to unladen moves. Finally, if some trucks remain unallocated after all counter values (NL_c, NE_c, NW_c) are reduced to zero, the action allocated to these trucks is to wait.

At first sight, it seems promising to consider K scenarios rather than a single one in order to generate a decision that is more resilient to the variability of future realizations. Yet, the consensus procedure has its own drawbacks. In fact, the final plan is created according to an aggregate-disaggregate method. This process might destroy the structure or consistency of the solutions that are optimal for different scenarios $k = 1, \ldots, K$. Therefore, the consensus solution might not produce a better combination of decisions than any single scenario-based one, i.e., $\theta_1, ..., \theta_K$.

*5.2.2 Restricted Expectation algorithm: RE**

In order to avoid the drawback of *Cs*, the Restricted Expectation algorithm RE^* is based on the selection of one single solution θ_k associated with a calibration scenario $k \in \{1, ..., K\}$. As for the consensus algorithm, we denote by θ_k the optimal values of the decision variables for the current period t and we denote by v_k the objective function value over the rolling horizon $\{t, ..., t+H\}$, for each calibration scenario $k = 1, ..., K$. In order to evaluate its quality, each decision θ_k is applied in period t for every remaining scenario $k' = 1, ..., K$, $k' \neq k$. (This is always feasible as all scenarios coincide in period t.) Then, we solve again model (6)-(11) for the H-period scenario constrained by the actions θ_k at t, and which coincides with scenario k' over $\{t+1, ..., t+H\}$. This yields an optimal value $v_{(k,k')}$, for all $k' \neq k$. So, we get the cumulated objective value Θ_k of action θ_k in the cross-evaluation procedure:

$$\Theta_k = v_k + \sum_{k'=1,\ k' \neq k}^{K} v_{(k,k')}. \tag{24}$$

Finally, the solution θ_k with the highest cumulated value $\Theta_k = \Theta_k^*$ is selected and actions θ_k are implemented in period t.

The cross-evaluation procedure is an attempt to evaluate the expected value of each solution, if it should be effectively applied. The qualifier "Restricted" reminds us that the number of calibrating scenarios K is much lower than the number of possible scenarios. When K increases, our confidence in the assumption that the cross-evaluation procedure provides a good estimate of the expected value of any solution θ_k increases. Note that, whereas *Cs* requires solving K optimization sub-problems at every decision period, RE^* requires solving K^2 such problems. This might become prohibitive in practice when K is large, even if each single optimization sub-problem is polynomially solvable.

5.3 Subtree algorithm: TR

Finally, in order to avoid the drawbacks of *Cs* (solution deconstruction) and RE^* (selection of a solution associated with a single scenario), we can use the problem formulation presented in Section 4.4 to compute a best-compromise solution over a subset of ST randomly generated scenarios. We refer to this

method as a *subtree algorithm*. Under the assumption that each scenario s is independently and identically generated with probability W_s (see Equation (3)), the objective function in Equation (12) provides a so-called *Monte Carlo estimator* of the expected profit due to the decisions made at the current period t (up to a constant factor ST). Monte Carlo estimators are commonly used in sampling-based optimization methods such as the generic Sample Average Approximation (SAA) approach; see, e.g., the discussion in [13].

Non-anticipativity constraints perform a similar role as the cross-evaluation procedure from RE^*, which fixes θ_k for the current period while allowing different decisions in the remaining part of the rolling horizon according to each scenario (see [22]). Whereas these constraints cannot ensure that the best decision has been taken for any single scenario, a compromise solution based on the highest average profit over the subtree can be selected. The influence of parameter ST for the numerical performance of the subtree is analyzed in Section 8.1 and Section 8.3.

6 INSTANCES

Temporal data: In our experiments, similarly to [28], the rolling horizon parameter H is set to 4 periods. In practice, these 4 periods, together with the current decision period t, may represent 5 days of information available for planning over the weekly horizon $\{t, \ldots, t+4\}$. The deterministically revealed part of the horizon RH contains 1 period beyond the current period t, so $H - RH = 3$ periods contain stochastic forecasts. The whole horizon WH involves 20 periods with pick-up periods $a_j \in \{1, \ldots, 20\}$, allowing deliveries outside the horizon. (This is consistent with the network flow formulation.) The rolling horizon process generates 16 effective decision steps, from period 1 to period 16, so that the length of the rolling horizon for the last decision is $H = 4$, as for all the other decision periods. An initial period "0" is added so as to start with a fleet of unladen trucks that might be initially moving or waiting. These starting conditions are similar for all algorithms and only slightly modify the expected profit value per period. Finally, the maximum unladen distance D is set to 1. Thus, no shortest path problems need to be solved in a preliminary phase (see Section 3).

a	b	Decision periods	c	d	e		
0	1	...	16	17	18	19	20

Fig. 2: Planning horizon (a: repositioning period for unladen fleet; b: first decision period; c: last decision period; c and d: deterministically revealed information periods for the last rolling horizon; e: stochastic information periods for the last rolling horizon)

Spatial data: In our experiments, cities are positioned on three grids of sizes 10, 20, 25; see Figure 3. They represent a permanent network of logistical platforms aggregating the local clients' orders. All edges are of length 1 and the driving speed is fixed so that it takes one period to cross an edge. The maximal distance between cities is 4. In the literature, it is usual to avoid uniform spatial distributions of clients (see Solomon or Li-Lim instances at *http://www.sintef.no/Projectweb/TOP*). Our tests were accordingly performed with complete graphs where each node of the grids is a city, but also with subgraphs where either 15 or 20 randomly selected nodes of the 25-city grid are designated as cities. So, in total, tests were performed with 9 types of graphs, namely, three complete grids of 10, 20 or 25 cities, three subgraphs (A, B or C) of 15 cities, and three subgraphs (A, B or C) of 20 cities.

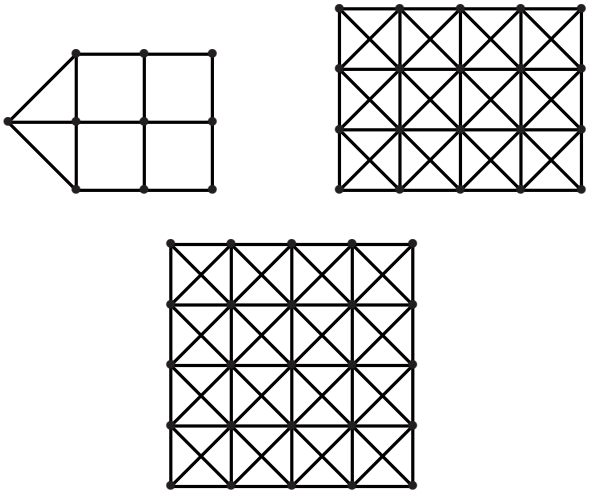

Fig. 3: Grids of 10, 20 and 25 cities

Orders: The data describing the orders are generated in two different ways. For a first set of instances (*orders linked to duration*), we assume that each set of cities is homogeneous. So, each order is randomly and uniformly assigned a pick-up city α_j, a destination β_j, and a pick-up date $a_j \in \{1, \ldots, WH\}$. The probability p_j that order j becomes available for transportation is assumed to depend on the trip duration. Namely, four probability distributions are considered, as displayed in Table 3. For example, according to distribution 2, the probability to become available is $p_j = 0.5$ for an order which has a trip duration equal to 3. The choice of these distributions is derived from the gravity model transposed in Reilly's law of retail gravitation (see [21]). Notice that p_j may be equal to 1, meaning that some deterministic information may be included in $\{t + RH + 1, \ldots, t + H\}$.

For each combination of a graph and a probability distribution, we generate an instance including 150 (potential) orders with pick-up dates in the horizon $\{1, \ldots, WH\}$, and another one including 200 (potential)

orders. Thus, we obtain 72 instances in the class '*orders linked to duration*' (9 graphs, 4 probability distributions, 2 densities of orders).

Table 3: Probability p_j of order availability according to the trip duration d_j: distributions 1 to 4

Duration	1	2	3	4
Distribution 1	1	0.5	0.33	0.25
Distribution 2	0.25	0.33	0.5	1
Distribution 3	0.75	0.5	0.33	0.25
Distribution 4	0.5	0.5	0.5	0.5

For a second set of instances (*orders linked to city range*), the size of each city is assumed to fall in one of three ranges, namely, large, medium, or small. The medium cities are most abundant in instances of type A, all city sizes are equally frequent in type B, and the small cities are most abundant in type C, as shown in Table 4. For example in the subgraph 15-25 A, there are 4 large cities, 6 medium cities, and 5 small cities. (Note that we did not include the complete grids in this set of instances.)

Table 4: Number of cities per range in each subgraph

City range	Large	Medium	Small
Subgraph 15-25 A: cities per range	4	6	5
Subgraph 15-25 B: cities per range	5	5	5
Subgraph 15-25 C: cities per range	2	4	9
Subgraph 20-25 A: cities per range	5	8	7
Subgraph 20-25 B: cities per range	6	7	7
Subgraph 20-25 C: cities per range	3	5	12

Then, α_j and β_j are randomly generated with probability proportional to 4, 3, and 2 for large, medium, and small cities, respectively. (This is again inspired from [21].) So, the larger the city, the more orders are picked up from or delivered to it. The pick-up dates a_j are uniformly drawn in the horizon $\{1,\ldots,WH\}$. For the probabilities p_j we consider again 4 possible distributions which depend on the range of the pick-up city a_j as shown in Table 5. For example, according to distribution 5, for an order which must be picked in a large city, the probability to become available is p_j = 0.75.

Each instance includes 150 (potential) orders. So, there are 24 instances in the class '*orders linked to city range*' (6 graphs, 4 probability distributions, 150 orders).

Table 5: Probability p_j of order availability according to the range of the pick-up city α_j: distributions 5 to 8

City range	Large	Medium	Small
Distribution 5	0.75	0.5	0.33
Distribution 6	1	0.5	0.25
Distribution 7	0.25	0.33	0.75
Distribution 8	0.5	0.5	0.5

Economic data: The profit per period for load j, that is, g_j, is set randomly and uniformly in the interval [80, 120] (monetary units: MU); the unladen trip cost per period, e, is set to 100 MU; the waiting cost per period, f, is set to 75 MU, as fuel cost usually represents approximately 25% of the total cost e. The difference between f and e (25 MU) is smaller than g_j, so that moving unladen towards an order is more profitable than staying at standstill.

Trucks: Each instance involves 10 trucks randomly and uniformly allocated to any city $\gamma_i(0) \in C$ in the initial period $t = 0$, for $i = 1,\ldots, 10$.

7 INTERPRETATION OF RESULTS

7.1 Bounds

In general, the policy selected by any algorithm suffers from two drawbacks when compared to the optimal solution π^* that could be computed if we had complete deterministic knowledge of the complete scenario over a fully revealed horizon of length WH. Firstly, it results from a sequential process implementing, at each decision period, actions issued from solutions over rolling subhorizons of limited length; but such a sequence of optimal or suboptimal short-term solutions does not build the optimal long-term one. Secondly, each rolling horizon solution is based on stochastic information and not on the fully revealed deterministic information over the rolling horizon. So, the policy might not be optimal even in a short-term perspective.

We denote by O^* the value of the (a posteriori, omniscient) optimal solution over the fully revealed complete horizon of length WH. This value can be computed by the deterministic network flow formulation of Section 4.2. It provides an upper bound for the best possible attainable value: $\mu_{\pi^*} \leq O^*$. We can easily compute this value over the relatively short horizon of $WH = 20$ periods.

Further, we denote by $O^*(H)$, the value of the solution obtained by solving optimally (a posteriori) a sequence of deterministic problems over fully revealed rolling subhorizons of length H. This value provides a very optimistic estimate of the value of policy π^*: that is, we cannot be sure that $\mu_{\pi^*} \leq O^*(H)$, always, but in practice, we can expect the inequality to hold, except for some pathological instances (as in [6]). Note that the difference $O^* - O^*(H)$ provides an estimate of the foregone profit due to the limited size H of the rolling horizon imposed by the order booking process and by the forecasting possibilities (see Section 7.2).

Finally, we denote by $O^*(RH)$ the value of the *myopic policy* derived by solving a sequence of deterministic problems over sub-horizons of size RH (which contain fully revealed information, by definition). In a sense, this policy is opposite to the optimistic policy where

all q_j are considered equal to 1 for $a_j \in \{t + RH + 1, ..., t + H\}$. The value $O^*(RH)$ provides an empirical lower bound on the best policy value μ_{π^*}. Indeed, π^* takes into account the same deterministic information as the myopic policy, as well as additional stochastic information over the rolling horizon. If $\mu_{\pi^*} \leq O^*(RH)$, this means that we draw no profit from the algorithm developed to deal with this additional information available to π^*.

In summary, the objective of our algorithms is to compute a best policy π^*, and we can expect that:

$$O^*(RH) \leq \mu_{\pi^*} \leq O^*(H) \leq O^*. \tag{25}$$

Note again that $O^*(H)$ and O^* are not derived from policies, since their computation relies on information that it not available at time t. Note also that O^* is always an upper bound on the other values in (25), but for any particular instance, it may happen that $O^*(H) < O^*(RH)$, and μ_π may turn out to be smaller than $O^*(RH)$ even for a policy π which takes stochastic information into account. Such "reversals of inequalities" are even more likely to be observed when we compare the values obtained by several algorithms on a particular scenario, or subset of scenarios. (see, for instance, the results in Table 11).

7.2 Value of the stochastic information

If $O^*(RH)$ is approximately equal to $O^*(H)$, i.e., if $O^*(RH)$ is not significantly smaller than $O^*(H)$ (see Section 7.3), it means that any reasonable algorithm should generate a profit μ_π which is close to both the lower bound and the upper bound. When this happens, the stochastic information contained in the horizon $\{t + RH + 1, ..., t + H\}$ appears to be useless, and the optimization process can be based on the deterministic information contained in the horizon $\{t, ..., t + RH\}$. Therefore, a preliminary check in our research framework is to analyze whether the gap between the bounds $O^*(RH)$ and $O^*(H)$ is economically significant. We accordingly define the metric EVMPM as:

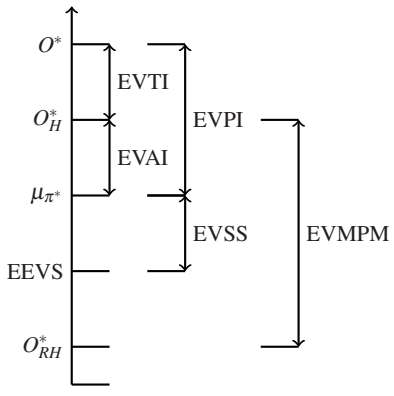

Fig.4: Bounds and values of information with expected value scenario

- *Expected value of the multi-period model*:
  $EVMPM = O^*(H) - O^*(RH)$ (26)

When EVMPM is significant, the best policy value returned by any of our algorithms provides an estimate of μ_{π^*} and we can compute the following values (EVSS should not be confused with EEVS, which has been defined in Section 5.1.3).

- *Expected value of the stochastic solution*:
  $EVSS = \mu_{\pi^*} - \text{EEVS}$ (27)
- *Expected value of the perfect information*:
  $EVPI = O^* - \mu_{\pi^*}$ (28)

These metrics are classical in stochastic optimization (see, e.g., [5]). Since we use them here in a multi-period framework, we sometimes find it instructive to complement them with additional metrics which split the value of the perfect information (EVPI) into two sub-values, namely: the value of the short-term accessible information, or EVAI, and the value of the long-term information pertaining to the tail, or EVTI, as introduced in [3]:

- *Expected value of the accessible information*:
  $EVAI = O_*(H) - \mu_{\pi^*}$ (29)
- *Expected value of the tail information*:
  $EVTI = O_* - O_*(H)$ (30)

It follows immediately that EVPI = EVAI + EVTI. See Figure 4) for an illustration of the relation between the different metrics. If some of the above values are statistically significant and economically relevant, it might be interesting for the transportation company to modify the information it collects by:

1) investing in its information system or changing its processes so as to collect more deterministic data over a longer horizon RH;
2) changing the booking process or forecasting tool in order to increase H.

Of course, beside improving the data collection, another option may also consist in developing optimization techniques leading to a better policy π^*.

7.3 Statistical validation of policy performance

Evaluating the expected value μ_π of a policy π theoretically requires to consider all potential realizations of the stochastic parameters, i.e., all potential scenarios. Since the number of scenarios is huge, μ_π can only be approximated on a restricted subset of scenarios. In our computational experiments, we evaluate μ_π by generating a random sample F of *test scenarios*. An estimate $\hat{\mu}_\pi$ of μ_π is then given by:

$$\hat{\mu}_\pi = \frac{1}{|F|} \sum_{s \in F} C(\pi, s), \tag{31}$$

where $C(\pi, s)$ is the value of the solution generated by π over scenario s. (Note that the test scenarios are conceptually and numerically distinct from the

calibration scenarios introduced in Section 5.2: the test scenarios are used to simulate the realization of random client requests, and to test *ex post* the performance of the algorithm in this simulated setting, whereas the calibration scenarios are gimmicks used by the algorithms in order to evaluate *ex ante* the impact of potential decisions.)

The bounds presented in Section 7.1 can be similarly approximated, leading to estimates $\hat{O}^*(RH)$, $\hat{O}^*(H)$ and $\hat{O}^*$. Analogously to (25), we expect to find:

$$\hat{O}^*(RH) \leq \hat{\mu}_{\pi^*} \leq \hat{O}^*(H) \leq \hat{O}^*, \tag{32}$$

For the sake of simplicity, we use the same notations for the estimates and for the true bounds in the rest of the document, i.e., O^* represents its estimate $\hat{O}^*$, and so forth.

As small differences between the estimated values $\mu_{\pi(1)}$ and $\mu_{\pi(2)}$ obtained for two distinct policies $\pi(1)$ and $\pi(2)$ might be due to random effects, their statistical significance must be assessed. For each fixed instance, we use the same set of test scenarios F in order to reduce the variance of the observed differences between policies. After checking normality of the distribution of results (using a Shapiro-Wilk non-normality test), we can apply a paired-sample Z-test to compare the values $\mu_{\pi(1)}$ and $\mu_{\pi(2)}$. One-sided tests are used in the discussions of our results.

$$H_0 : \mu_{\pi(1)} = \mu_{\pi(2)} \; vs. \; H_1 : \mu_{\pi(1)} > \mu_{\pi(2)}. \tag{33}$$

As the objective function is profit maximization, we say for short that policy $\pi(1)$ *outclasses* policy $\pi(2)$ on a given instance if we can reject H_0 *vs.* H_1 at a predefined confidence level. In our experiments, the number of test scenarios used to evaluate μ_π, for every policy π and for every instance, is $|F| = 30$ (meaning that $16 \times 30 = 480$ single-period decisions are actually generated by each algorithm for each instance), and the confidence level is fixed at 95%; that is, policy $\pi(1)$ outclasses policy $\pi(2)$ if the Z statistic is larger than 1.65.

8 EXPERIMENTAL RESULTS

8.1 Implementation and algorithmic parameters

All algorithms have been implemented in Java. The experiments have been performed on a personal laptop computer (Core 2 Duo 2GHz, 2GB of RAM, Windows). The integer linear programming problems were solved using IBM ILOG CPLEX 12 with default settings. The running time per decision period for a single scenario was around 0.1 sec, but the data management operations (inputs-outputs) linked to the rolling horizon lead to a few seconds of computing time per decision period. So, clearly, solving a single period problem, as required on a daily basis in a real-world environment, is computationally affordable. However, for the purpose of statistical validation and due to the rolling horizon process, we had to tackle 16 decision periods per test scenario and $|F| = 30$ test scenarios per instance.This leads to 480 single-period network flow computations per instance for each of the three bounds $O^*(RH)$, $O^*(H)$ and O^*, and for each algorithm *Opt, Mod* and *EG*.

After preliminary testing, the algorithmic parameter K representing the number of calibrating scenarios for the consensus algorithm *Cs* was set to 10. Indeed, initial analysis showed that most of the decisions over the K calibrating scenarios are similar. This leads to identical decisions for loading, moving unladen or waiting over most of the calibration scenarios, and decision frequencies expressed as percentages lie either between 0 and 10%, or between 90 and 100% when considering all calibrating scenarios. Therefore, increasing K has little influence, as also demonstrated by a sensitivity analysis on this algorithmic parameter. With $K = 10$, *Cs* requires solving $480 * K = 4800$ network flows models per instance in order to test and to validate the performance of the algorithm (but only $K = 10$ flow models to make a decision in any time period).

For the restricted expectation algorithm RE^*, the running speed becomes an issue as K^2 optimization subproblems have to be solved at each period. Sensitivity analysis also showed that increasing K beyond 10 calibrating scenarios does not statistically improve the values obtained. This results in $480 * K^2 = 48000$ network flow computations per instance for RE^* (and again, only $K^2 = 100$ flow models at any single decision period).

For the subtree algorithm TR, the number of calibrating scenarios ST was set either to 10 or 30, leading to subtree models labeled TR_{10} and TR_{30}. In our experiments, the running time increased almost proportionally to the number of calibration scenarios (about 1 second per decision period for TR_{10}, and about 3 seconds for TR_{30}). So, even if the data management time was larger due to the model size, the total running time for the subtree algorithm TR was smaller than for the multiple-scenario algorithms RE^*.

8.2 Result tables

Results and statistical comparisons of policy performances are provided in Tables 6 to 8, where instances are referenced in the form "Distribution-Graph"; so, for example, instance 1-20-25C is associated with the probability distribution 1 from Table 3 and with the subgraph 20-25C from Table 4. The left half of Tables 6, 7, 8 displays the value of the objective function produced by various algorithms for three sets of instances (average value over 30 test scenarios, for each instance). More precisely, the (average) value of $O^*(RH) = O^*1$ and of $O^*(H) = O^*4$ is displayed for each instance. The difference EVMPM $= O^*4 - O^*1$ is the expected value of the

Table 6: 150 orders and availability linked to duration

Info Alg. Inst.	Policy performance in percentage										Z-statistic value							
	O^*16	**LB** O^*1 0%	*Opt*	*Mod*	**EEVS** *EG*	Cs	RE^*	TR_{10}	TR_{30}	**UB** O^*4 100%	O^*4 > O^*1	O^*16 > O^*4	O^*4 > EG	O^*4 > TR_{10}	TR_{30} > TR_{10}	O^*4 > TR_{30}	TR_{30} > EG	TR_{30} > π^*
1-10	120.4	3413	22.8	17.3	37.3	31.2	12.4	48.6	**58.2**	4151	9.48	4.39	9.71	5.74	1.63	5.55	**3.52**	
1-15-25A	153.0	1962	12.9	38.8	38.4	51.4	43.1	65.7	**70.2**	2998	9.52	7.91	8.48	3.88	0.64	3.92	**4.12**	
1-15-25B	153.8	1878	13.7	44.7	49.2	45.5	26.7	66.5	**75.5**	2598	9.24	8.36	4.52	3.21	**1.71**	2.50	**3.10**	
1-15-25C	176.0	1977	32.8	43.5	67.1	45.2	36.9	72.8	**85.2**	2712	6.12	8.15	2.80	2.16	1.04	1.23	1.64	
1-20	135.0	1948	14.8	41.3	52.5	38.9	46.5	69.8	**71.0**	2764	9.64	7.13	7.96	6.40	0.17	5.16	**1.98**	
1-20-25A	167.8	1096	6.8	32.5	62.4	21.3	44.9	73.1	**78.1**	1838	9.99	7.82	3.39	3.71	**4.13**	3.35	**1.71**	
1-20-25B	149.6	168	23.3	41.0	46.2	42.0	31.8	**70.6**	60.0	1275	7.46	5.49	6.39	3.38	-1.54	4.48	1.43	-1.54 (TR_{10})
1-20-25C	199.8	1489	-22.1	30.1	24.1	27.0	-24.7	61.5	**67.9**	2102	7.07	6.05	5.43	2.29	0.78	1.99	**4.25**	
1-25	164.9	1558	-83.6	6.5	7.9	12.6	-32.1	**54.9**	50.6	2145	5.89	5.70	7.47	3.84	-0.50	4.59	**4.88**	-0.50 (TR_{10})
2-10	163.7	-541	18.4	38.9	44.7	37.7	26.8	67.4	**74.3**	-96	6.33	8.89	4.87	3.17	0.78	2.26	**2.16**	
2-15-25A	221.3	2792	69.2	70.2	65.8	70.9	63.7	**77.2**	76.4	4336	9.39	12.52	3.53	2.69	-0.20	2.42	1.56	-0.20 (TR_{10})
2-15-25B	186.1	2538	65.1	66.3	70.3	51.0	62.4	83.2	**87.4**	3250	5.35	6.21	1.94	1.86	0.37	1.08	1.27	
2-15-25C	136.6	1889	36.7	60.4	67.5	73.1	42.5	78.3	**82.4**	2770	9.49	4.90	5.85	4.91	0.72	3.12	**2.50**	
2-20	204.6	1850	59.6	**74.5**	57.7	53.0	39.3	71.6	70.1	2648	6.16	8.60	2.72	2.76	-0.16	2.69	0.99	-0.36 (Mod)
2-20-25A	190.1	1917	51.6	71.1	81.1	69.4	60.2	82.7	**83.3**	2914	8.41	9.02	2.10	2.40	0.10	2.40	0.24	
2-20-25B	150.9	4320	30.4	40.0	54.5	57.4	53.2	**77.5**	74.2	5269	7.86	8.36	4.92	3.46	-0.65	3.72	**1.86**	-0.65 (TR_{10})
2-20-25C	180.9	1545	65.2	86.5	87.1	79.6	62.0	86.3	**89.2**	2341	6.59	6.94	1.31	1.46	0.43	1.26	0.25	
2-25	167.3	4950	11.4	50.0	65.0	64.2	42.6	**69.8**	61.0	5664	7.36	5.86	3.54	2.37	-0.92	3.43	-0.45	-0.92 (TR_{10})
3-10	142.1	3074	-7.5	47.5	32.1	37.9	20.7	62.9	**63.9**	3601	9.01	7.46	5.30	3.76	0.09	3.21	**2.35**	
3-15-25A	198.2	266	-8.4	50.7	33.3	57.1	29.3	68.0	**83.0**	917	8.23	8.56	5.60	2.59	1.60	1.30	**3.71**	
3-15-25B	249.2	178	27.9	28.7	39.5	30.8	-13.2	91.8	**92.6**	565	3.60	6.49	2.59	0.35	0.06	0.39	**2.99**	
3-15-25C	155.4	-98	-5.3	17.3	24.8	31.2	6.6	51.1	**61.6**	624	8.52	6.01	6.17	4.21	1.24	4.18	**3.04**	
3-20	154.1	236	9.0	31.5	33.1	38.9	32.7	56.0	**60.6**	855	7.89	5.61	3.95	3.37	0.59	3.25	**2.47**	
3-20-25A	180.7	1236	-37.6	11.2	25.8	11.8	-22.6	39.0	**41.7**	1913	7.96	6.88	6.04	5.95	0.23	4.98	1.41	
3-20-25B	199.2	1200	-33.2	-9.5	26.7	16.8	2.9	54.8	**69.1**	1759	6.15	10.85	4.79	2.87	1.38	2.07	**2.22**	
3-20-25C	164.1	93	20.4	54.4	40.2	37.1	42.3	**61.8**	53.0	1166	9.56	8.62	6.84	5.01	-1.24	5.75	1.35	-1.24 (TR_{10})
3-25	177.4	-572	-1.3	43.7	22.8	28.7	15.7	61.5	**63.4**	184	7.76	8.96	5.72	3.28	0.20	4.00	**4.06**	
4-10	121.5	3359	51.4	51.4	60.4	**68.0**	45.5	64.8	63.9	4313	8.54	5.67	7.06	6.97	-0.19	5.32	0.60	-0.60 (Cs)
4-15-25A	201.6	2395	24.2	24.2	49.1	38.0	26.7	60.1	**79.1**	2889	5.19	6.28	3.71	2.65	**2.08**	1.54	**2.34**	
4-15-25B	155.8	3638	25.5	25.5	48.5	53.3	25.5	64.2	**69.0**	4683	8.43	7.70	5.39	4.87	0.55	3.39	**2.56**	
4-15-25C	162.2	3136	-19.0	-19.0	27.8	28.7	-9.4	52.1	**55.1**	3840	7.72	6.67	6.24	5.18	0.32	5.11	**2.54**	
4-20	165.0	2483	-14.5	-14.5	27.6	30.2	7.2	55.6	**63.1**	3275	7.67	7.68	9.98	4.26	0.79	4.65	**4.42**	
4-20-25A	145.1	2590	10.0	10.0	34.8	41.0	12.9	56.6	**66.0**	3536	6.81	6.27	7.72	4.41	1.45	3.88	**3.64**	
4-20-25B	142.8	2289	53.9	53.9	52.0	62.7	42.3	68.7	**72.4**	3491	10.75	6.98	6.39	3.49	0.82	3.73	**3.38**	
4-20-25C	177.8	2344	7.6	7.6	69.5	48.6	14.5	65.4	**70.1**	3185	6.89	8.32	3.07	3.66	0.54	3.15	0.06	
4-25	148.0	3027	15.1	15.1	36.3	23.1	10.0	48.6	**52.1**	3985	9.19	6.99	6.83	6.41	0.48	5.13	**1.85**	
Average	168.4	1877	15.2	35.6	**46.2**	43.2	25.8	65.6	**69.3**	2679								

multi-period model. In view of Eq. (32), algorithms that make effective use of the stochastic information should perform somewhere between the bounds O^*1 and O^*4: accordingly, in Tables 6, 7, 8, the performance of each policy is expressed as the percentage of EVMPM closed by the policy. In other words, for policy π^*, Tables 6, 7, 8 report the value

$$\frac{\mu_{\pi^*} - O^*(1)}{O^*(4) - O^*(1)} = \frac{\mu_{\pi^*} - O^*(1)}{EVMPM},$$

expressed in percentage points. (Note that O^*16 approximates the optimal value O^* obtained when perfect information is revealed over the whole horizon, and hence O^*16 is usually larger than 100%).

This presentation allows us to measure how each policy fills the gap between myopic optimization and the policy that would benefit from deterministic information over each rolling horizon. The right half of Tables 6, 7, 8 displays the value of the Z-test statistic which allows us to check the pairwise relative performance of algorithms.

Table 9 displays the performance metrics for the results in Tables 6 to 8 and also for the larger instance results in Table 10. The EVMPM is provided in absolute value and μ_{π^*} is displayed, so that their economic significance can be checked and information values compared to them.

8.3 Results for 150-order instances with availability linked to duration

Let us first consider the results in Table 6. Before we discuss the relative performance of the algorithms, we can note that the average value of O^*4 is roughly 1.43 (= 2679/1877) times higher than the average O^*1, meaning that the expected value of the multi-period model, EVMPM, is economically significant (see Table 9). Hence, there is value to be gained by taking the stochastic information into account.

Next, we observe that neither the probability distribution, nor the graph type or range appear to be discriminating parameters for this class of instances. The subtree algorithm with 30 calibrating scenarios TR_{30} is best on 28 instances out of 36. The subtree

Table 7: 200 orders and availability linked to duration

	Policy performance in percentage						Z-statistic value					
Info		**LB**	**EEVS**			**UB**	O^*4	O^*16	O^*4	O^*4	TR_{30}	TR_{30}
Alg.	O^*16	O^*1	EG	Cs	TR_{30}	O^*4	>	>	>	>	>	>
Inst.		0%				100%	O^*1	O^*4	EG	TR_{30}	EG	π^*
1-10	119.1	4938	46.5	55.1	**69.8**	5974	15.47	6.47	10.25	6.23	**3.86**	
1-15-25A	164.2	3859	47.0	41.7	**63.1**	5017	7.12	9.09	5.24	5.04	**1.91**	
1-15-25B	149.9	4295	29.6	23.1	**47.5**	5128	8.70	5.53	7.51	3.75	1.18	
1-15-25C	156.5	4417	26.7	31.2	**56.5**	5186	8.07	9.39	6.92	4.13	**2.83**	
1-20	165.4	4655	29.6	56.1	**79.2**	5294	6.13	8.73	5.68	1.98	**3.95**	
1-20-25A	138.4	3186	33.1	28.0	**53.8**	4311	11.05	8.62	7.83	8.11	**2.57**	
1-20-25B	147.8	4220	17.3	18.2	**50.1**	5178	7.89	8.15	7.77	5.16	**3.09**	
1-20-25C	139.5	4074	42.6	24.2	**54.8**	4903	6.32	7.45	4.87	4.50	1.23	
1-25	162.1	3444	34.7	29.3	**62.3**	4148	5.88	7.06	3.79	4.32	**1.89**	
2-10	136.4	4945	46.0	44.5	**70.7**	5638	7.06	4.88	4.86	3.24	**2.78**	
2-15-25A	197.1	5803	54.1	**87.8**	79.9	7217	7.78	10.85	4.55	2.62	**2.61**	-1.06 (Cs)
2-15-25B	174.7	5712	65.0	47.1	**68.5**	6660	11.13	11.95	2.92	3.10	0.37	
2-15-25C	190.3	6152	48.3	44.5	**68.0**	6898	6.54	8.78	3.71	2.42	**2.01**	
2-20	145.7	3851	70.7	83.1	**90.0**	4973	9.04	5.95	3.69	1.35	**3.39**	
2-20-25A	165.6	6611	43.3	40.4	**62.5**	7435	6.65	7.14	5.61	3.12	**2.08**	
2-20-25B	188.1	6569	53.0	70.9	**84.3**	7558	8.17	9.00	4.31	2.34	**4.10**	
2-20-25C	204.9	5190	69.3	64.1	**86.8**	6001	7.03	10.72	3.17	1.26	1.64	
2-25	164.4	4538	36.1	36.7	**48.6**	5334	7.04	7.43	6.97	5.37	1.50	
3-10	140.5	5369	48.6	43.8	**66.4**	6045	9.11	7.63	4.98	4.23	**1.81**	
3-15-25A	164.1	2862	32.5	50.3	**73.9**	3694	7.06	6.22	5.87	3.73	**3.64**	
3-15-25B	160.6	3048	53.2	55.6	**68.9**	4315	8.47	9.17	5.28	3.61	**2.13**	
3-15-25C	172.1	3904	43.0	*10.8*	**59.0**	4657	8.17	7.58	5.57	4.66	1.61	
3-20	127.9	3076	30.4	26.4	**52.2**	4000	8.89	4.59	6.59	7.26	**2.34**	
3-20-25A	157.9	3906	21.1	26.9	**43.4**	4876	8.60	6.26	9.18	5.94	**1.90**	
3-20-25B	153.7	2810	29.9	36.9	**52.7**	3711	9.21	6.44	7.87	5.24	**1.89**	
3-20-25C	172.5	1599	64.8	68.1	**82.6**	2711	7.99	6.10	3.24	2.03	**2.32**	
3-25	167.8	3107	25.4	16.8	**47.3**	3986	6.01	7.08	7.09	5.43	**2.11**	
4-10	135.4	5669	61.6	67.1	**79.0**	6571	8.95	5.58	4.05	2.34	**2.54**	
4-15-25A	204.3	6352	3.9	*-10.4*	**29.6**	6866	4.82	6.84	5.10	3.76	**1.88**	
4-15-25B	204.6	3618	57.5	**68.5**	65.8	4738	10.93	9.58	4.93	3.35	0.98	-0.30 (Cs)
4-15-25C	136.5	4723	43.9	55.7	**63.3**	5817	9.94	6.37	6.53	4.21	**2.38**	
4-20	193.6	4334	54.3	52.8	**71.1**	5206	6.86	9.76	3.45	2.91	1.46	
4-20-25A	175.7	5185	46.8	50.0	**70.3**	6030	8.03	8.61	4.62	3.16	**2.54**	
4-20-25B	171.7	5876	43.3	43.6	**78.2**	6806	8.68	6.62	5.89	2.27	**3.48**	
4-20-25C	180.3	3569	36.8	*5.6*	**62.4**	4193	6.17	7.62	4.43	3.24	**1.70**	
4-25	220.1	5514	**59.9**	11.9	51.4	6079	3.64	11.19	2.30	3.02	-0.73	**-0.73**(EG)
Average	165.3	4472	**43.0**	41.8	**64.3**	5365						

Table 8: 150 orders and availability linked to city range

	Policy performance in percentage						Z-statistic value					
Info Alg. Inst.	O^*16	**LB** O^*1 0%	**EEVS** EG	Cs	TR_{30}	**UB** O^*4 100%	O^*4 > O^*1	O^*16 > O^*4	O^*4 > EG	O^*4 > TR_{30}	TR_{30} > EG	TR_{30} > π^*
5-15-25 A	222.0	760	73.6	**80.0**	79.2	2486	8.91	12.72	3.14	2.40	0.78	-0.10 (Cs)
6-15-25 A	156.1	2348	78.6	**90.8**	89.7	4725	9.29	9.65	3.98	1.87	**2.41**	-0.27 (Cs)
7-15-25 A	171.0	350	57.2	68.0	**70.7**	930	6.00	5.25	3.61	3.09	1.39	
8-15-25 A	187.3	2120	**54.3**	13.8	53.4	2720	4.57	5.02	3.01	3.79	-0.08	**-0.08** (EG)
5-15-25 B	153.1	3511	57.7	61.2	**81.6**	4410	10.06	5.54	4.40	2.15	**3.74**	
6-15-25 B	165.7	5957	55.8	42.8	**60.3**	6797	6.78	7.34	5.00	3.96	0.50	
7-15-25 B	194.7	-242	56.5	60.4	**61.0**	402	6.25	7.70	3.77	3.18	0.49	
8-15-25 B	201.4	820	86.7	60.8	**100.0**	1553	8.37	9.84	1.15	0.00	**1.81**	
5-15-25 C	192.4	599	64.1	53.8	**78.8**	1231	7.87	7.37	3.58	1.65	1.39	
6-15-25 C	125.9	3196	62.7	78.3	**88.0**	5027	10.18	7.16	6.40	3.80	**4.26**	
7-15-25 C	179.2	2852	63.9	49.6	**70.4**	3524	6.95	9.22	3.98	2.97	0.73	
8-15-25 C	192.0	4395	47.0	20.0	**63.5**	5156	5.99	9.12	4.27	3.59	1.44	
5-20-25 A	195.1	3139	63.9	45.2	**65.9**	4052	7.69	10.11	3.87	3.30	0.22	
6-20-25 A	153.8	4126	52.1	54.4	**74.3**	5658	10.46	8.99	4.33	3.32	**2.77**	
7-20-25 A	253.9	298	38.6	32.1	**44.5**	760	4.47	6.82	3.61	3.11	0.48	
8-20-25 A	225.7	3292	7.3	-36.5	**21.9**	3742	4.00	5.74	4.46	3.57	0.62	
5-20-25 B	141.9	2573	62.9	33.2	**68.4**	3395	6.39	4.27	3.13	2.78	0.75	
6-20-25 B	147.4	4226	62.7	53.4	**74.2**	4952	8.12	7.11	5.54	3.05	1.43	
7-20-25 B	176.7	1859	52.1	52.7	**66.1**	2752	7.98	9.37	4.64	3.16	1.32	
8-20-25 B	165.1	2597	49.8	25.6	**54.2**	3575	9.60	8.88	5.90	5.42	0.51	
5-20-25 C	171.7	1278	51.4	61.2	**67.7**	2084	7.00	7.68	4.92	4.10	**1.99**	
6-20-25 C	215.3	-52	39.1	23.6	**56.1**	582	7.53	9.85	5.67	3.52	**1.66**	
7-20-25 C	142.9	2543	53.6	54.0	**61.3**	3447	5.80	5.13	3.72	3.51	0.97	
8-20-25 C	150.3	1088	67.3	41.7	**71.3**	2024	7.91	7.64	4.42	4.04	0.59	
Average	178.4	2235	**56.6**	46.7	**67.6**	3166						

algorithm with 10 calibrating scenarios TR_{10} is best 6 times out of 36, and is often close to TR_{30}. On average, TR_{30} closes 69.3% of the EVMPM (and in view of the overall good performance of TR_{30}, μ_{π^*} is only slightly better: it closes 70.6% of the EVMPM gap). In fact, the Z-values in Table 6 indicate that the subtree algorithm TR_{30} is never outclassed by any algorithm, not even by the best policy π^* found for any instance.

TR_{30} outclasses TR_{10} three times ($Z > 1.65$ in bold in the sixth column), but it is never outclassed by TR_{10}.

A deeper performance analysis on the algorithmic parameter ST was also performed by setting $ST = 50$. This allowed us to conclude that the larger the subtree, the better TR_{ST} numerically performs on average while reducing the standard deviation. Yet, as TR_{30} is never statistically outclassed by TR_{50}, numerical experiments were eventually restricted to $ST = 30$ calibrating scenarios to reduce running time for the validation phase. Note that in a real-world setting, as each optimization step is performed only daily, TR_{50} might be used for each decision period.

The expected value scenario algorithm EG, which gives the expected value of the expected value solution (EEVS), yields on average the second best policy after TR_{ST} (it closes 46.2% of the EVMPM). Yet, its performance varies a lot. The expected value of the stochastic solution, i.e., EVSS = μ_{π^*}-EEVS, is 24.4% of the EVMPM, where μ_{π^*} is estimated as the best value obtained by any algorithm and is displayed in bold for each instance. Table 6 shows that TR_{30} outclasses EG 24 times ($Z > 1.65$ in bold in the penultimate column), meaning that using the stochastic information and a dedicated algorithm is relevant both theoretically and economically, as the profit increase represents 23.1% of a significant EVMPM.

The consensus algorithm Cs is the third best policy, but far behind TR_{ST}. Algorithms Opt, Mod and RE^* usually yield poor results and sometimes even underperform when compared with the myopic bound O^*1. The weak performance of RE^*, in particular, was confirmed on other test instances (not reported here); it is quite disappointing in view of the relative sophistication of this algorithm which was shown to deliver very good results in other problem settings (see, e.g., [3], [30]).

From Table 9, one can notice that the expected value of perfect information over all instances is high: EVPI = $O^*16 - \mu_{\pi^*} = 97.8\%$ on average. O^*16 is always significantly larger than O^*4, and the expected value of the tail information is high: EVTI = $O^*16 - O^*4$ = 68.4% on average leading on average to a value of the accessible information (EVAI) of 29.4%.

8.4 Results for 200-order instances with availability linked to duration

Based on the previous observations, we decided to restrict our subsequent tests to TR_{ST}, EG and Cs. From Table 7, one can observe that the EVMPM is still large (20% of O^*1). Here again, neither the probability distribution, nor the graph type or range appear to be discriminating parameters for the instances with 200 orders and availability linked to

duration. The subtree algorithm TR_{30} is still the best policy, it closes the EVMPM gap by 64.3% on average as compared to μ_{π^*} which closes 64.8% of EVMPM (see Table 9).This confirms that the expected value of the accessible information (EVAI) is relatively high, at 35.2% on average. The expected value of the perfect and tail information (EVPI = 100.5%, EVTI = 65.3%) and the expected value of the stochastic solution (EVSS = 21.8%) are similar to those observed in Section 8.3 and remain high on average. The subtree algorithm TR_{30} is beaten by the consensus algorithm Cs twice and by the expected value scenario algorithm EG once, but it is never statistically outclassed. On the other hand, TR_{30} outclasses EG 27 times out of 36, meaning that developing a stochastic policy pays off in most cases.

8.5 Results for 150-order instances with availability linked to the pick-up city range

The results are quite similar for this class of instances. Looking at Table 8, one can notice that the EVMPM is still large, at 41.7% of O^*1. Neither the probability distribution, nor the graph type or range appear to be discriminating parameters for these instances. The subtree algorithm TR_{30} provides the best value except for three instances where it is marginally beaten by Cs (twice) or by EG (once). On average, TR_{30} closes about 67.6% of the gap (as compared to 67.7% for μ_{π^*}). From Table 9, we confirm that the expected value of the perfect information (EVPI), of the accessible information (EVAI) and of the tail information (EVTI) remain high on average, at 110.6%, 32.3% and 78.4%, respectively. The expected value scenario algorithm EG performs better than previously, only 11% behind TR_{30} and so, the expected value of the stochastic solution (EVSS) decreases (11.1%). However, the subtree algorithm TR_{30} is never outclassed, but outclasses 7 times EG.

8.6 Larger instances

To validate the computational efficiency of the algorithms (see Section 4.5), larger instances involving 25 trucks and 350 orders transported over a grid of 25 cities were also tested, with 4 probability distributions linked to trip duration. Results obtained for these instances are shown in Table 10. In all cases, the optimal integer solution of the subtree model is directly obtained by solving its linear relaxation, so that computation times remain within a few seconds (9 seconds on average) for the solution of the model at every period t. It is important to remember that in an industrial implementation of the method, only this computing time is relevant, since only one subtree model will be solved on a daily basis.

The same instances with a rolling horizon of length 15 and larger subtrees (algorithm TR_{50}) have also been tested successfully, with similar conclusions regarding the tractability of the optimization models at every period. Finally, very large instances of TR_{30} including 100 trucks and, respectively, 100 or 120 loads (on average) per period have been tested. The running time for solving each model is, on average, 55 seconds or 143 seconds, respectively, which remains again very short for practical purposes.

Table 10:
350 orders and availability linked to duration

	Policy performance in percentage					
Info Inst. Alg.	O^*16	**LB** O^*1 0%	**EEVS** EG	Cs	TR_{30}	**UB** O^*4 100%
1-25	129.9	9712	49.1	44.4	**78.0**	11696
2-25	145.5	15366	74.5	70.2	**81.9**	17857
3-25	133.8	8350	35.2	20.7	**76.7**	9961
4-25	135.0	11902	42.0	25.5	**71.0**	13604
Average	136.1	11332	**50.2**	40.2	**76.9**	13279

8.7 Performance analysis

From the previous results, we can conclude that TR_{30}, the subtree algorithm including 30 calibrating scenarios, usually performs best among our algorithms. It outclasses all other algorithms (Opt, Mod, Cs, RE^*), whereas it is not outclassed itself on any of the instances that we tested. The EVMPM is economically relevant for all instance classes, and TR_{30} closes approximately two-thirds of this gap, or even three-fourths on average for larger instances. Other tests also show that TR_{30} always outclasses O^*1, proving the interest of using the information from the rolling horizon. Unfortunately, the value provided by TR_{30} is almost always significantly smaller than the upper bound O^*4. This means that the stochastic information is useful but that we are not able to use it to equal the performance of an a posteriori solution. So, the expected value of the accessible information (EVAI) remains significant. Similarly, by observing the gap between algorithms based either on fully revealed information over the rolling horizon (O^*4), or on fully revealed information

Table 9: Performance metrics for Tables 6, 7, 8 and 10
(columns 3 – 8 are expressed as percentages of the EVMPM)

	EVMPM	$\mu_{\pi^*} - O^*1$	**EEVS**$-O^*1$	**EVSS**	**EVPI**	**EVAI**	**EVTI**
150 orders and availability linked to duration	801	70.6%	46.2%	24.4%	97.8%	29.4%	64.8%
200 orders and availability linked to duration	894	64.8%	43.0%	21.8%	100.5%	35.2%	65.3%
150 orders and availability linked to city range	931	67.7%	56.6%	11.1%	110.6%	32.3%	78.4%
350 orders and availability linked to duration	1947	76.9%	50.2%	26.7%	59.2%	23.1%	36.1%

over the whole horizon (O^*16), we see that the tail information (EVTI) is quite high, but decreases when instances become larger. One can conclude that increasing the rolling horizon length might be useful. But in practice, this length may be constrained by the order-booking process and may not be increased on demand. The expected value scenario algorithm EG performs well. Nevertheless, EG is often outclassed by the subtree algorithm TR. So, the expected value of the stochastic solution (EVSS) is high, meaning that using the stochastic information based on the probability distribution is profitable. Finally, the results show that the expected value of the perfect information (EVPI= $O^*16-\mu_{\pi^*}$) is high. Thus, there may be some room for improvement of the algorithms, as the value of the best policy we obtain for the stochastic problem appears to be significantly smaller than the value of the optimal solution for the fully revealed deterministic problem.

8.8 Robustness analysis

Except Opt, which radically sets all random variables q_j to 1 (meaning that Opt does not depend on the probability distributions at all), all other algorithms are based on a presupposed knowledge of the distribution parameters p_j, i.e., on the exact availability probability of each order. This is a strong assumption. In practice, it may be hard to determine p_j, as its value can only be derived from the aggregation of historical data or from expert opinions. Therefore, it is important to analyze the sensitivity of the results to the exact valuation of the distributions. We call this process "robustness analysis" in the present framework.

In order to carry out this analysis, tests were performed when the probabilities p_j used to generate the test scenarios (which simulate the "real" observations) are different from the probabilities p'_j used to generate the calibration scenarios or the subtrees (which model the presupposed distributions). The instances under consideration involve 150 orders. They are named in the format "p-Graph-s". "Graph" denotes one of two subgraphs, namely, either 15-25A or 20-25A. The suffix s indicates whether the pick-up cities α_j and destination cities β_j are generated uniformly ($s=u$) or depending on the city range ($s=r$) (see Section 6). The prefix p indicates the real probability distribution used for the test scenarios: $p_j=p$ for all j, where p takes one of the values 0.2, 0.5, or 0.8. The notation $EG_{0.5}$ refers to the expected value scenario algorithm calibrated with a distribution where $p'_j=0.5$ for all j. The notation TR_{30}^{X} refers to the subtree algorithm $TR30$ calibrated with $p'_j=X$, where X is either 0.2, 0.5, or 0.8, but can be (very) different from p. Results are shown in Table 11.

Numerically, no calibration of TR_{30}^{X} is robust against severe miscalibrations of the distributions, but any TR_{30}^{X} still provides better results than a myopic policy in most cases. As one should expect, the general trend is that better solutions are obtained when the calibration parameter X gets closer to the "real" parameter p. In particular, $TR_{30}^{0.5}$ is reasonably robust against underestimation ($p=0.8$) or overestimation ($p=0.2$) of the availability of orders. (This is to be contrasted with $EG^{0.5}$ which always performs poorly.) When the probability p increases in the test scenarios, the performance of TR_{30}^{X} improves and may approach the upper bound O^*4 when X is sufficiently close to p (in particular, when $p=0.8$). Generally speaking, however, the value returned by TR_{30}^{X} remains low (close to the lower bound O^*1) when p is very small, suggesting that these instances may be hard to solve. So, the expected value of the accessible information (EVAI) tends to increase when p decreases.

The statistical comparison of policy performance in Table 11 confirms that $TR_{30}^{0.5}$ is never outclassed. Moreover, $TR_{30}^{0.5}$ often outclasses $TR_{30}^{0.2}$ and $TR_{30}^{0.8}$ when $p=0.5$. Logically, $TR_{30}^{0.2}$ or $TR_{30}^{0.8}$ are often outclassed when the real distribution is opposite to the distribution that they use for calibration.

As a conclusion, when accurate information is lacking, it may be recommended in practice to approximate the real probability distribution by assuming that $p_j=0.5$.

Table 11: Robustness with respect to probability distribution

	Policy performance in percentage					Z-statistic value								
Info Alg. Inst.	O^*16	**EEVS** $EG^{0.5}$	$TR_{30}^{0.2}$	$TR_{30}^{0.5}$	$TR_{30}^{0.8}$	O^*4 > O^*1	O^*16 > O^*4	O^*4 > $EG^{0.5}$	O^*4 > $TR_{30}^{0.2}$	O^*4 > $TR_{30}^{0.5}$	O^*4 > $TR_{30}^{0.8}$	$TR_{30}^{0.2}$ > $TR_{30}^{0.5}$	$TR_{30}^{0.2}$ > $TR_{30}^{0.8}$	$TR_{30}^{0.5}$ > $TR_{30}^{0.8}$
0.2-15-25 A-u	361.4	25.2	61.9	**65.0**	38.2	3.87	4.40	2.17	1.31	1.09	1.60	-0.12	0.6	0.80
0.2-15-25 A-r	283.7	34.5	**87.2**	72.5	-0.1	4.34	6.20	3.94	1.07	2.10	5.60	1.26	5.14	**3.82**
0.2-20-25 A-u	229.0	31.9	**70.5**	45.3	35.6	5.68	5.49	4.26	2.28	3.40	3.89	1.64	1.97	0.68
0.2-20-25 A-r	298.4	3.7	-7.0	**9.7**	6.7	3.57	5.57	3.23	4.32	3.20	3.89	-0.59	-0.55	0.10
Average: $p=0.2$	293.1	23.8	**53.2**	48.1	20.1									
0.5-15-25 A-u	201.6	49.1	48.0	**79.1**	59.2	5.19	6.28	3.71	3.26	1.54	2.71	**-2.36**	-0.80	**1.90**
0.5-15-25 A-r	187.3	**54.3**	21.9	53.4	35.8	4.57	5.02	3.01	5.16	3.79	4.34	**-2.31**	-0.74	1.34
0.5-20-25 A-u	145.1	34.8	47.5	**66.0**	43.3	6.81	6.27	7.72	5.65	3.88	5.13	**-2.50**	0.43	**2.94**
0.5-20-25 A-r	225.7	7.3	10.3	**21.9**	-17.6	4.00	5.74	4.46	3.71	3.57	6.28	-0.62	1.07	**1.93**
Average: $p=0.5$	189.9	36.4	31.9	**55.1**	30.2									
0.8-15-25 A-u	152.6	91.0	71.4	**111.2**	106.1	11.62	9.03	1.03	3.83	-1.25	-0.97	**-6.65**	-5.06	0.78
0.8-15-25 A-r	217.0	44.4	43.1	87.1	**87.4**	4.80	8.58	3.28	3.46	0.87	1.04	**-2.68**	-2.91	-0.02
0.8-20-25 A-u	129.7	85.3	47.5	96.1	**99.3**	14.44	6.06	2.59	8.73	1.04	0.15	**-8.92**	-8.01	-0.86
0.8-20-25 A-r	184.4	20.8	39.2	45.4	**47.2**	5.64	9.25	4.51	4.69	3.45	4.70	-0.42	-0.55	-0.14
Average: $p=0.8$	170.9	60.4	50.3	84.9	**85.0**									

9 CONCLUSIONS

Even though they have been investigated for a long time, multi-period stochastic planning problems remain extremely hard to solve. In managerial practice, therefore, such problems are often reduced to a sequence of mono-period deterministic problems, so as to avoid the difficulties linked to the formulation of complex models and the implementation of sophisticated solution methods. Such pragmatic approaches, however, frequently lead to strongly suboptimal decisions.

In this paper, we have discussed a multi-period stochastic vehicle allocation problem arising in transportation planning. We have proposed several heuristics to deal with this problem in a rolling-horizon setting where information is gradually revealed to the decision maker. A main objective of our research was to establish that (relatively) simple algorithms can deliver significant improvements over myopic approaches. The most successful algorithms explicitly take into account the multi-period framework, by modeling the (deterministic or stochastic) information which is available within a rolling horizon of limited length. Since the estimation of probability distributions may prove excessively difficult and may act as a deterrent to implementation in some industrial settings, we have tested the robustness of our algorithms against misspecification of the distributions. Specific conclusions drawn from our experiments are as follows:

- For the instances that we have tested, the expected value of the multi-period model appears to be quite large: EVMPM lies roughly between 20% and 40% of O^*1 meaning that it is potentially profitable to model the (stochastic) information contained in the rolling horizon $\{1, ..., H\}$, rather than restricting the attention to the (fully revealed, deterministic) information in the shorter horizon $\{1, ..., RH\}$. This is in line with the observations of [28], but the conclusions extend here to the stochastic version of the problem.
- In fact, the best value produced by our algorithms (denoted μ_{π^*}) closes about two thirds of the EVMPM and outclasses, in particular, the value EEVS produced by a simple deterministic approximation (based on the expected value scenario). This translates into an economically attractive expected value of the stochastic solution (EVSS = EEVS – μ_{π^*}) ranging roughly from 11% to 27% (see Table 9). Since hoping to close the full gap is unrealistic, due to the presence of uncertain orders in the subhorizon $\{RH+1, ..., H\}$, we conclude again that the performance of the algorithms is attractive from the practitioners' point of view.
- For the stochastic version of our DAVP model, an integer programming formulation based on subtrees of scenarios turns out to produce the best results. In our experiments, including a rather small number of scenarios in the subtree already allows us to compute solutions which, in the rolling horizon setting, significantly outperform alternative policies derived from a single scenario, or from combinatorial heuristics based on multiple scenarios (consensus or restricted expectation algorithms).
- The integer programming formulation of the subtree model is computationally tractable and can be solved in a few seconds by a commercial solver, which makes it potentially attractive for industrial implementations. Moreover, when the probability distribution of the availability of orders is difficult to estimate (which is likely to be the case in practice, especially when the number of orders is very large), even a rough model of the stochastic component (e.g., setting $p_j = 0.5$ for all j) proves sufficiently robust to provide significant improvements over the myopic approach which does not take forecasts into account.

ACKNOWLEDGEMENTS

We thank Yasemin Arda, Frédéric Semet and two anonymous reviewers for their helpful comments. This project was partially funded by the Belgian Scientific Policy Office (Grant P7/36).

REFERENCES

1. Ahuja RK, Magnanti TL, Orlin JB (1993) Network Flows: Theory, Algorithms and Applications, Prentice-Hall, Upper Saddle River, NJ
2. Angelelli E, Bianchessi N, Mansini R, Speranza MG (2009) Short term strategies for a dynamic multiperiod routing problem. Transportation Research Part C: Emerging Technologies 17(2):106–119
3. Arda Y, Crama Y, Kronus D, Pironet T, Van Hentenryck P (2014) Multi-period vehicle loading with stochastic release dates. EURO Journal on Transportation and Logistics 3:93–119
4. Berbeglia G, Cordeau JF, Laporte G (2010) Dynamic pickup and delivery problems. European Journal of Operational Research 202(1):8–15
5. Birge JR, Louveaux F (1997) Introduction to Stochastic Programming. Springer, Berlin
6. Blackburn JD, Millen RA (1980) Heuristic lot-sizing performance in a rolling-schedule environment. Decision Sciences 11(4):691–701
7. Crainic TG (2003) Long-haul freight transportation, Kluwer Academic Publishers, Boston, MA, pp 451–516. International Series in Operations Research and Management Science
8. Dejax P, Crainic T (1987) Survey paper – a review of empty flows and fleet management models in freight transportation. Transportation Science 21(4):227–248
9. Desrosiers J, Dumas Y, Solomon MM, SoumisF (1995) Handbooks in Operations Research and Management Science, vol 8, Elsevier Science B.V., chap 2, pp 35–139
10. Eurostat (2018) Road freight transport by journey characteristics. URL http://ec.europa.eu/eurostat/statistics-explained/index.php?titel=Road_freight_transport_by_journey_characteristics§oldid=377506
11. Frantzeskakis L, Powell W (1990) A successive linear approximation procedure for stochastic dynamic vehicle allocation problems. Transportation Science 24(1):40–57
12. Keskinocak P, Tayur S (1998) Scheduling of timeshared jet aircraft. Transportation Science 32(3):277–294
13. Homem-de Mello T, Bayraksan G (2014) Monte Carlo sampling-based methods for stochastic optimization. Surveys in Operations Research and Management Science 19(1):56 – 85
14. Meyers CA, Schulz AS (2009) Integer equal flows. Operations Research Letters 37(4):245–249
15. Mitrovic` Minic` S, Krishnamurti R, Laporte G (2004) Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transportation Research Part B: Methodological 38(8):669–685
16. Powell WB (1986) A stochastic model of the dynamic vehicle allocation problem. Transportation Science 20(2):117–129
17. Powell WB (1988) A comparative review of alternative algorithms for the dynamic vehicle allocation problem, Elsevier Science Publishers, Amsterdam, pp 249–291
18. Powell WB (1996) A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. Transportation Science 30(3):195–219
19. Powell WB, Towns MT, Marar A (2000) On the value of optimal myopic solutions for dynamic routing and scheduling problems in the presence of user noncompliance. Transportation Science 34(1):67–85
20. Ramaekers K, Caris A, Maes T, Janssens G (2015) Pickup and delivery selection: Problem formulation and extension to problem variants. Information Technology and Management Science 18:84–90
21. Reilly WJ (1931) The Law of Retail Gravitation. Reilly, W. J., New York
22. Ruszczyński A, Shapiro A (2003) Handbooks in Operations Research and Management Science, vol 10, Elsevier Science B.V., chap 1, pp 1–64
23. Savelsbergh MWP, Sol M (1995) The general pickup and delivery problem. Transportation Science 29(1):17–29
24. Schönberger J (2005) Operational Freight Carrier Planning. Springer, Berlin Heidelberg
25. Secomandi N (2000) A rollout policy for the vehicle routing problem with stochastic demands. Operations Research 49(5):706–802
26. Sethi S, Sorger G (1991) A theory of rolling horizon decision making. Annals of Operations Research 29:387–416
27. Spivey MZ, Powell WB (2004) The dynamic assignment problem. Transportation Science 38(4):399–419
28. Tjokroamidjojo D, Kutanoglu E, Taylor GD (2006) Quantifying the value of advance load information in truckload trucking. Transportation Research Part E: Logistics and Transportation Review 42(4):340–357
29. Topaloglu H, Powell WB (2007) Sensitivity analysis of a dynamic fleet management model using approximate dynamic programming. Operations Research 55(2):319–331
30. Van Hentenryck P, Bent RW (2006) Online Stochastic Combinatorial Optimization. MIT Press, Cambridge, Massachussets
31. Yang J, Jaillet P, Mahmassani HS (2004) Real-time multivehicle truckload pickup and delivery problems. Transportation Science 38(2):135–148
32. Zolfagharinia H, Haughton M (2014) The benefit of advance load information for truckload carriers. Transportation Research Part E: Logistics and Transportation Review 70:34–54