Multi-modal transport of perishable products with demand uncertainty and empty repositioning: A scenario-based rolling horizon framework

M. SteadieSeifi, N. Dellaert *, T. Van Woensel

School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands

ABSTRACT

We study the planning of a multi-modal transportation system with perishable products, demand uncertainty and repositioning of the empty Returnable Transport Items (RTIs). We propose a rolling horizon framework where we periodically re-optimize. As such, relevant responses and actions to new occurred demand are taken, and possible updates to the transportation and repositioning plans can be made. Our rolling horizon framework considers the uncertainty of customer demand, formulated as a Scenario-based Two-Stage Program (STSP) for which a set of scenarios is generated. An Adaptive Large Neighborhood Search (ALNS) algorithm is used to solve this scenario-based problem. Our proposed ALNS algorithm employs new operators and strategies to solve this complex and large problem. We give detailed computational analysis on the properties of our framework, evaluating the effects of stochastic demand, and we provide practical insights.

1. Introduction

Perishable supply chains are international businesses and the horticultural industry of the Netherlands is a well-known example. Everyday, around 25 million cut flowers and plants, with an average daily turnover of more than 18 million Euros (FloraHolland, 2017) are transported from around the world to the auction houses in the Netherlands, to get auctioned, sold, and further transported throughout Europe and beyond. Large numbers of products are handled and moved everyday. During execution, controlling such a system involves regularly observing the current status, comparing actual with planned operations, and making new and/or repair decisions in response to the new state of the system.

In SteadieSeifi et al. (2017), we studied the tactical planning of the long-haul transportation of perishable products with repositioning of Reusable Transport Items (RTIs). We presented a mode-space-time network and modeled it as an extension to the classic Fixed-charge Capacitated Multi-commodity Network Flow Problem (FCMNP). In this setting, we assumed that product demand was deterministic. For perishable products, however, the product demand is usually non-stationary. There are daily, weekly (weekday versus weekend), seasonal (summer versus winter), and promotional trends influencing the volume of products that a customer orders throughout the year. These trends necessitate regular (e.g. daily or weekly) process control to improve the operational efficiency of the related perishable supply chains.

If the demand realizations are less than their expected values, the planned fleet is sufficient to transport the products and reposition the empty RTIs. However, when the actual demand exceeds the planned capacity, companies resort to outsourcing the operations or renting extra cool transport resources which are usually more expensive. We argue that these trade-offs and their associated costs should be included in the planning process. In this paper, we take this uncertainty of customer demand into account, by considering demand realization in time buckets within a rolling horizon setting. As such, we acknowledge that customer demands are uncertain, and a deterministic flow and repositioning plan might cause capacity shortage in the planned multi-modal fleet. This influences the transport of products directly, or indirectly, by a delayed repositioning of empty RTIs. Alternatively, by considering future demand, consolidation opportunities in future time intervals can be considered, potentially resulting in more efficient transportation.

Following Powell et al. (1995) and Pillac et al. (2013), we employ a time-based planning having fixed time intervals (e.g. every 12 h), at the end of which new decisions are made. We have a planning horizon of a few days, which is rolled after a certain time interval (e.g. after every 12 h), denoted as roll. This approach is called rolling horizon approach. Rolling horizon approaches are very useful in handling large-scale or dynamic problems. The only drawback of using a rolling horizon framework is the possibility of ending up with sub-optimal solutions. Since the size of our problem and its practical details is huge, finding a...
good solution in reasonable time is much more important than finding the optimal solution (Pillac et al., 2013).

Similar to the approach presented in Pillac et al. (2013), we re-optimize the static problem considering the actual data and system state information at the end of each roll. Re-optimization can be partial or complete depending upon the complexity of the underlying decision problem. In long-haul transportation of perishable products, consolidation is the key factor in decreasing operational costs, therefore, in this paper, we use a complete re-optimization setup. A complete re-optimization includes all non-finished customer orders and is allowed to change all parts of the system state as long as they are not yet fixed (e.g. the vehicles have not started their transport). Limited re-optimization can be used to keep the number of changes during the planning horizon small.

In our rolling horizon approach (Fig. 1), at the end of each roll, we update the system and collect all new data revealed in the roll $\phi$. Then, we solve a re-optimization problem for a finite number of future time periods. The new plan is executed until the end of the next roll. The collected data includes the set of new orders arrived during the roll, as well as the state of the system and resource availability by the end of roll.

The periodically solved re-optimization problem is a roll-dependent version of the planning problem of SteadieSeifi et al. (2017) with uncertain demand. In this paper, we use stochastic programming to deal with demand uncertainty (Powell et al., 1995). In this approach, the problem is optimized over the known and predicted demand, taking the expected costs of responding to various demand outcomes (scenarios) into account. It is extended from the model of SteadieSeifi et al. (2017) and will be solved by the proposed ALNS algorithms.

The contributions of this paper are threefold:

1. We study the long-haul multi-modal transportation of perishable products with repositioning of Reusable Transport Items (RTIs), considering non-stationary stochastic demand.
2. The problem we study in this paper is a dynamic extension to the planning problem of SteadieSeifi et al. (2017). This is the first time a rolling horizon approach is designed for operational planning of the multi-modal transportation for perishable products with repositioning of empty RTIs.
3. We add to the literature by proposing a rolling horizon framework where we use a Scenario-based Two-stage Stochastic Program (STSP) to deal with uncertainty of demand. In this STSP, a set of scenarios is generated and an extension of the ALNS algorithm of SteadieSeifi et al. (2017) is used to solve this scenario-based problem.

The outline of this paper is as follows. Section 2 gives an overview on the relevant literature. In Section 3 then, we describe this planning problem and its uncertain nature, and in Section 4, we explain the proposed rolling horizon framework and its properties, and take a look at different modeling of stochastic customer demand. We also present the two-stage stochastic program that we solve in our rolling horizon framework. In Section 5, we describe our extended ALNS algorithm, and in Section 6, we present a comprehensive computational analysis, and finally, in Section 7, we provide some concluding remarks and describe potential future work.

2. Related literature

First, we describe a number of rolling horizon approaches in transportation applications, and afterwards, we describe possible solution approaches.

For dynamic VRPs, Chen et al. (2006); Schönberger (2010); Yang et al. (2004); Mitrović-Minič et al. (2004) adopt a rolling horizon approach to their planning problems. In these examples, the decision timing and the time intervals are fixed, and decisions are made at the given moment in time.

Rolling horizons are often applied to reduce the problem size in situations where the demand is deterministic. For instance, Choong et al. (2002) present a Mixed-Integer Program (MIP) for a real-world empty container management system, and analyze the effect of length of the planning horizon on the quality of distribution plans and their outsourcing. Rakke et al. (2011) model an annual planning inventory routing of liquefied natural gas as a MIP. The problem size is too big, therefore, they solve its sub-problems via a rolling horizon heuristic. Examples of a rolling horizon approach in a stochastic environment are for instance Erera et al. (2005), who present a MIP for the operational planning of a multi-modal tank container where they integrate planning the repositioning of containers with their routing. Bandeira et al. (2009) who model an integrated empty and loaded container transshipment system and Bock (2010), who introduces a real-time-oriented control approach for efficient consolidation and transshipment and dynamic handling of disturbances such as vehicle breakdown and accidents. Furthermore, Ahmt et al. (2015) considers container positioning at a terminal and Cordeau et al. (2015) embed a routing problem with heterogeneous fleet into a rolling horizon algorithm in order to incorporate the dynamics of new requests and the loading capacity of the fleet. Close to our setting, van Heeswijk et al. (2016) study a real-time multi-modal transportation planning, where routes have to be found for each arriving order. Operational planning of long-haul transportation of perishable products with repositioning of RTIs, can be placed as a special class of dynamic and stochastic transportation planning literature. The most important and frequently used approaches for modeling and solving dynamic and stochastic problems are (1) simulation-optimization,
rolling horizon procedures, and 3) dynamic programming (Powell, 2009). An example of simulation optimization is Bock (2010), who designs a metaheuristic algorithm based on a Variable Neighborhood Search (VNS) which applies dynamic neighborhood search for generating new plans. The rolling horizon procedures are often MIP-based, such as Ahm et al. (2015), who present a MIP model and several valid inequalities for a container positioning at a terminal and then solve the large instances of this problem with a rolling horizon based heuristic. Also heuristics are used here, such as van Heeswijk et al. (2016). For their multi-modal planning problem a “best k-route” generating heuristic is used to find routes for each arriving order, and a consolidation heuristic to find the best re-consolidation option.

A portion of the literature is dedicated to the application of Approximate Dynamic Programming (ADP) in resource allocation and management. ADP was developed to tackle the curse of dimensionality in solving stochastic and dynamic problems. Interested readers are referred to Topaloglu and Powell (2006, 2007, 2005); Lam et al. (2007); Bouzaiene-Ayari et al. (2014). They apply ADP to an empty container allocation problem arising in a sea-cargo industry. They first model a simple two-port two-voyage system and extend it to multi-port multi-voyage network. Bouzaiene-Ayari et al. (2014) propose an ADP framework for a rail fleet sizing and routing problem, with a streamlined single commodity model, a more detailed multi-commodity model, and a multi-attribute model.

The application of rolling horizon methods with stochastic elements is limited. An example is Di Francesco et al. (2013). They study the effect of partial or complete port disruption in empty container repositioning in a liner shipping system. They model it as a time-space representation and consider a set of different disruption scenarios. They also include some non-anticipatory conditions to equalize the here-and-now decision variables over all scenarios.

3. Problem description

The long-haul transportation of perishable products with repositioning of empty RTIs is an extension of the Fixed-charge Capacitated Multi-commodity Network Flow Problem (FCMNP), where the main decisions are the flow of products, the repositioning of empty RTIs, the selection of transport modes and schedules to do these jobs, and the number of vehicles needed for each transport mode.

Reusable Transport Items (RTIs), being the loading units used for transportation, are the key elements of this problem, and the main decisions are defined on their flow throughout the network. Their number is limited and their flow is subject to strict resource balance constraints.

The physical transportation network is characterized by $i = 1, \ldots, N$ hub locations, and arcs $(i, j)$ representing different routes connecting these locations. Between each location pair, at least one transportation mode $m$ can operate. Each transport mode has its given schedules, and its vehicles operate based on them. Each mode has a specific temperature, which for example shows the temperature inside a truck trailer, a train car, or a barge storage room. In the horticultural industry, the product freshness is approximated by a Time Temperature Sum (TTS) measure, representing the total time that products can be transported in different temperature regimes. In this paper, a maximum allowed TTS (e.g. 200 h-degree) is enforced for each order, and this limitation is added to the model via a new set of constraints.

Vehicles of a transport mode are capacitated based on number of Reusable Transport Items (RTIs). RTIs can be small (e.g. cages) or medium (e.g. trolleys). There is a limited number of vehicles available for each transport mode. Each mode also has its own fixed cost per hour, fixed costs per vehicle, and variable costs per RTI per hour. The hub locations also have their own operation time, temperature, and holding cost per RTI per hour, which can again be different from one time interval to the next.

Demand is represented by orders. We consider two sets of orders: realized and predicted orders.

1. Realized orders arrive to the system at different time periods. A realized order is characterized by its pair of origin and destination locations, its volume, its pickup and delivery schedules, and its freshness requirements. Since RTIs are the loading units used to transport the products, the order volume is represented in the number of RTIs needed (e.g. 2 trolleys) to transport products from the origin to the destination. Products can be picked up and loaded onto the RTIs at an earliest given time, and should be delivered to the destination and unloaded at a latest given time. As long as the order is transported within the length of this schedule, its flow is feasible. Each order has a maximum TTS limit.

2. Predicted orders are generated based on historical information about the expected customer demand for products in future time intervals. Their locations, pickup and delivery time windows, their volume, and desired TTS, are subject to uncertainty. In Section 4.2, we explain in detail how this demand is generated.

The main decisions are related to the RTI flow (loaded or empty), repositioning, handling, and holding throughout the network and over the given planning horizon. Then, the role of updated decisions in a particular planning horizon is to look for opportunities to consolidate new RTI flows with the existing ones. Note that the fleet arrangement not only serves the known RTI flows, but is also prepared for anticipated ones. The main decisions are as follows:

- Flow of new customer demand revealed in previous time intervals,
- Flow of the anticipated customer demand in the coming time intervals,
- New and updated repositioning of Returnable Transport Items (RTIs) for both revealed and anticipated demand,
- New and updated selection of transport modes and schedules, which transport the revealed customer demand, while preparing for future demand fluctuations,
- New and updated number of vehicles (real or ad-hoc) needed for each of these transport modes.

In this paper, the objective function is to minimize total operational costs, including flow costs of the loaded RTIs, flow costs of the assigned and repositioned empty RTIs, locational costs, and finally, costs of using the modes, over all rolled planning horizons.

At each decision epoch, plans from previous decision moments might not have started yet. For example, the previous epoch, we planned to call a truck for tomorrow. This truck option is re-examined in the planning of the current epoch, in which case, it can be replaced with other cheaper modes or better consolidation options. However, once a mode started its execution, it continues to the planned destination without stopping or changes.

In the next section, we first describe the rolling horizon framework with the elements that are used in the mathematical model formulation.

4. Rolling horizon framework

The planning problem of our paper belongs to the time-based planning class, where there are fixed time intervals (e.g. every 12 h), at the end of which new decisions are made. A re-optimization procedure periodically solves the corresponding static problem (Pillac et al., 2013).

In practice, there is an infinite planning horizon, meaning that there is no ending time period or system state to be reached. However, revealed customer orders have delivery deadlines and repositioning operations do not take infinite hours. Moreover, logistics service providers usually reveal their available capacity and price for a certain time in the future (e.g. only for the next three months). Additionally, since all revealed orders and all repositioning jobs are done within a shorter period (e.g. two weeks), there is no need to stretch the planning horizon further. Consequently, we truncate our optimization problem into a smaller planning problem with a finite and short planning horizon $T$, which is then rolled after a certain time interval of $r$ (e.g. after every 12 h). $r$ hence denotes the length of a roll (Fig. 1).
At the end of a roll \( \varphi \) and after collecting new data, we make an anticipation of demand (via generating a number of scenarios) for future time intervals. Then, we formulate a Scenario-based Two-stage Stochastic Program (STSP) with time periods \( t = \varphi, \varphi + 1, \ldots, \varphi + T \) to optimize the truncated stochastic problem used in the rolling horizon framework. The objective function of the truncated problem is then to minimize total operational costs, including flow costs of the loaded RTIs, flow costs of the assigned and repositioned empty RTIs, locational costs, costs of using the modes, and finally, the costs of ad-hoc capacity increase, over the planning horizon \( T \).

In the remainder of this section, we describe the structure of our rolling horizon framework in detail. We start with defining the state of system and properties of a solution of the truncated problem we solve at the end of a roll. An Adaptive Large Neighborhood Search (ALNS) algorithm is used to solve the STSP.

### 4.1. The system state and a roll solution

The **system state** is a snapshot taken at the end of a roll \( \varphi \), representing a combination of the status of orders, RTI positions, status of routes and the fleet.

The set of orders keeps track of the status of the orders. At the end of roll \( \varphi \), an order can be: **new**, **not-planned**, or **planned**. If the deadline of not-planned orders passed, they are **lost**.

The set of RTI positions contains information on the location, availability timing, and volume for all RTIs. It is similar to an inventory concept indicating at which time period, how many RTIs are available at a location. This set is then used for assigning empty RTIs to orders when needed. If empty RTIs need to be repositioned, **empty orders** are generated and added to the set of all orders.

The set of routes represents the scheduled flow of (loaded or empty) RTIs. This set only depicts **planned orders**. A route at the end of roll \( \varphi \) has its own state: **not started**, **en-route to the destination**, **en-route to a transshipment**, and **completed**. A route is not started until its scheduled departure time fits in roll \( \varphi \). If a route is executed, its state changes to en-route. However, depending on whether the next location on the route is the destination or a transshipment location, its state is different. Routes to a transshipment can still be included in the **re-planning** strategy.

A route that has an **en-route to the destination** state is not included in the re-planning. A route is finally complete when the RTIs arrive to the destination location during roll \( \varphi \).

The **fleet** set tracks the multi-modal vehicles used for moving the flows. A member of the fleet can have either of both states: **used** or **unused**. Used fleet are the ones executing the routes with en-route states. The unused ones were planned but their departing time is beyond roll \( \varphi \).

To find future potential consolidation options, unused fleet are taken into account in the re-planning strategy.

A **roll solution** \( z_\varphi \) is the solution obtained for the roll \( \varphi \). It transforms into a different solution \( z_{\varphi+1} \) for roll \( \varphi + 1 \), given the system state. Completed routes and their relevant orders are removed from \( z_{\varphi+1} \). Only their statistics are kept in the history.

Fig. 2 gives an overview of the data flow and the decisions taken for each roll \( \varphi \).

### 4.2. Anticipation of future demand

At the end of each roll, orders \( p = 1, \ldots, \mathcal{P} \) are realized and all their volumes, \( \mathbf{w}_p \), are known. Orders of future days are unknown but predicted. Each predicted order \( p' \), is characterized by the random number \( \tilde{w}_{p'} \), and distribution function \( \Psi(\tilde{w}_{p'}, t) \), and average number \( \mu_{p'} \). This distribution is time \( t \) dependent, taking into account seasonal (weekly, daily, or even hourly) trends. However, from one day to another, and from one customer to another, they are independent.

The random demand volume for each \( p' \) is denoted as \( \tilde{w}_{p'} \). We analyzed the historical data of year 2013 that was available from Tosi (2014); Verhoeven (2014); Vlassak (2014); Rosenboom (2014), into a frequency distribution of daily demand volume for each customer \( p' \) (Rubin, 2010). These frequency distributions are used to represent all possibilities of \( \tilde{w}_{p'} \) and their probability \( \Psi(\tilde{w}_{p'}, t) \). It is important to note that the daily, weekly and seasonal trends are taken into account in generating

---

**Fig. 2.** Data flow and decisions for a roll \( \varphi \).
these distributions.

When the anticipated demand for future days is more than the total planned capacity, companies resort to outsourcing. This option and its associated costs should then be included into the planning. In this regard, we add an ad-hoc fleet capacity increase, denoted as $\theta_P$ and its associated cost per unit $C_P$ for $p = 1, \ldots, P$ to the model to reflect the amount of order volume that is outsourced.

### 4.3 Scenario generation

In operational planning problems where new plans and adjustments should be made in a really short time of a few minutes to a few hours, the size of the stochastic program becomes troublesome.

To obtain a manageable stochastic program, for each order set $p', \forall p'$, we generate a set of $\gamma = 1, \ldots, \Gamma$ scenarios with uniform probability of $\rho_{\gamma}$, and extend the deterministic MIP of SteadieSeif et al. (2017) to include these scenario realizations and transform it into a Scenario-based Stochastic Program (SSP). In the next section, we formulate this stochastic program and its two stages.

### 4.4 Scenario-based Two-Stage Program (STSP) with stochastic demand

In this section, we present the mathematical formulation for the STSP with stochastic demand of roll $\varphi$. Note that for simplification reasons, we remove the dimension of $\varphi$ representing a roll in the rolling horizon framework from the STSP formulation, but in fact, this model can have different parameter settings and number of variables in each roll of our rolling horizon explained later, depending on order sets $P$ and $P'$, and the available transportation modes.

Demand is represented by orders. A real order $p$ is characterized by the number $w_{p}$ of RTIs needed to load and transport products from an origin location $O(p)$ to a destination location $D(p)$. The products can be picked up and loaded onto the RTIs at an earliest time of $P_T(p)$ and should be delivered to the destination and unloaded at the latest time of $D_T(p)$. Product freshness preservation is evaluated by the Time Temperature Sum (TTS) measure, and for each order, it is assumed that the summation of these TTSs should be less than $C_{\text{TTS}}$ (say 200 h < degree).

Fleets of different transportation modes over the coming planning horizon, with given schedules, temperature, capacity, and number of available vehicles are available. In comparison to SteadieSeif et al. (2017) though, the characteristics of available fleet is roll-dependent, and from one roll to the next, might be different.

At the end of each day, we generate a set of $\gamma = 1, \ldots, \Gamma$ scenarios with uniform probability of $\rho_{\gamma}$. Each scenario is a set of predicted orders. A predicted order $p \in \{1, \ldots, P\}$ is distinguished by its random volume $w_{p}^{\gamma}$ with daily average of $\mu_{p}$ and a distribution function $\Psi(w_{p}^{\gamma})$. We add an ad-hoc fleet capacity increase, denoted as $\theta_{p}^{\gamma}$ for $p = 1, \ldots, P$ to the model to reflect the amount of order volume that is outsourced. We later add the associated cost per unit $C_P$ to the objective function.

The goal of this SSP is to find a flexible multi-modal fleet to answer all realizations of random demand volumes of predicted orders $p'$, as well as the known volume of orders $p$. Therefore, we reformulate the SSP into two stages where in the first stage, the transportation mode fleet is arranged and the RTI flows and repositioning of real orders are planned. In the second stage then, the arranged fleet with the remaining resources and capacities are used for transporting the products and repositioning the RTIs for future scenario orders.

The objective function of our operational planning problem and its embedded STSP is to minimize total operational costs, including flow costs of the loaded RTIs, flow costs of the assigned and repositioned empty RTIs, locational costs, and finally, costs of using the modes, over all rolled planning horizons. Note that there is no cancellation or change of path for the moving modes.

Assume $V$ is the set of all nodes in the mode-space-time network. Let $A_1 = \{(a_{i,j,l}, m_1, m_2) \in A(V \times V) | i \neq j, m_1 = m_2 \}$ be the set of all feasible and given travel arcs, and let $A_2 = \{(a_{i,j,l}, m_1, m_2) \in A(V \times V) | i = j, m_1 = H \}$ be the set of all feasible location arcs, representing loading, unloading, holding and waiting arcs. Similarly, let $A_3 = \{(a_{i,j,l}, m_1, m_2) \in A(V \times V) | m_1 = H \}$ be the set of all feasible arcs (traveling, loading, unloading, and waiting) in the network related to mode $H$. The length of a travel arc is $r_{i,j,l}^{m_1}$, representing the travel time between $i$ and $j$ by mode $m$, and the length of a location arc is $r_{i,j,l}^{m_0}$ representing the handling or transshipment time at location $l$.

For each order $p$ or $p'$, there are three flow decisions: laden, assign, and repos flow decisions, and there are two sets of variables defined for each of them: (1) $x_l^m$ to show how many RTIs enter an arc $(a_{i,j,l}, m_1, m_2)$, and (2) $x_r^m$ to show how many RTIs exit the arc. Likewise, there are two sets of variables $y$ and $\gamma$ to show the number of vehicles respectively entering and exiting an arc. The auxiliary binary variables $b$ and $\theta$ are also added to help calculating TTS of real and predicted orders. Three scenarios $\Gamma$ are also added to the model to keep track of the inbound and outbound flows at each location, and to connect the flows of loaded and empty RTIs throughout the network.

The roll-dependent STSP model with product quality preservation and empty RTI management is then formulated as follows:

### 4.4.1 First stage

Compared to a classic STSP, here, the first stage not only includes fleet arrangement decisions $y$ and $\gamma$, but also the decisions $x_l$ and $x_r$ on the RTI flow and repositioning of the real orders. No matter what scenario is realized for predicted orders, the real orders are planned in this stage. Therefore, their related decisions are placed in the first stage.

Note that repositioning decisions only become positive if RTIs are needed to be repositioning during the current planning horizon $t = \varphi, \varphi + 1, \ldots, \varphi + T$. In Section 6, we test an end-of-the-week RTIs repositioning rule where an inventory $S_t[\geq 0]$ of empty RTIs should be available at the inbound hub locations at beginning of each week. Assuming $t_{\text{end}}$ to be such a (repetitive) repositioning deadline, if $t_{\text{end}} > \varphi + T$, all repositioning decisions become zero.

\[
\begin{align*}
\min & \sum_{m_0}^{M} \sum_{p_0 \in P} \sum_{l \in L} \sum_{j \in J} \sum_{\gamma \in \Gamma} C_l^{m_0} \times r_{l,j}^{m_0,\gamma} \\
& + \sum_{a_{i,j,l} \in A_2} \sum_{p_0 \in P} \sum_{\gamma \in \Gamma} \left[ \sum_{m_0}^{M} C_{\text{TTS}} \times r_{l,j}^{m_0,\gamma} + \sum_{m_1}^{M} C_{\text{TTS}} \times r_{l,j}^{m_1,\gamma} \right] \\
& + \sum_{a_{i,j,l} \in A_2} \sum_{p_0 \in P} \sum_{\gamma \in \Gamma} \left[ \sum_{m_0}^{M} C_{\text{TTS}} \times r_{l,j}^{m_0,\gamma} + \sum_{m_1}^{M} C_{\text{TTS}} \times r_{l,j}^{m_1,\gamma} \right] \\
& + \sum_{a_{i,j,l} \in A_2} \sum_{p_0 \in P} \sum_{\gamma \in \Gamma} \left[ \sum_{m_0}^{M} C_{\text{TTS}} \times r_{l,j}^{m_0,\gamma} + \sum_{m_1}^{M} C_{\text{TTS}} \times r_{l,j}^{m_1,\gamma} \right] \\
& + \sum_{a_{i,j,l} \in A_2} \sum_{p_0 \in P} \sum_{\gamma \in \Gamma} \left[ \sum_{m_0}^{M} C_{\text{TTS}} \times r_{l,j}^{m_0,\gamma} + \sum_{m_1}^{M} C_{\text{TTS}} \times r_{l,j}^{m_1,\gamma} \right]
\end{align*}
\]
are added to help calculating TTS.

\[
\gamma_{\text{cat}}^{\text{cat}} \left( h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \right) = \gamma_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \quad \forall \text{cat} \in \{\text{laden, empty, repos}\}, \\
\begin{array}{c}
\text{a}_{i, o} \in A, \\
p = 1, \ldots, P
\end{array}
\]  
(2)

\[
\gamma_{\text{cat}}^{\text{cat}} \left( h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \right) = \gamma_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \quad \forall \text{cat} \in \{\text{laden, empty, repos}\}, \\
\begin{array}{c}
\text{a}_{i, o} \in A, \\
p = 1, \ldots, P
\end{array}
\]  
(3)

Constraints (8) are flow conservation constraints. These constraints define variables \( U \) as the total net flow (loaded RTIs, assigned empty and repositioned empty RTIs) for each order at each location and time period. These equalities imply that the total inbound flows to each node should be equal to its total outbound flows.

\[
U_{p, t}^{\text{cat}} = \sum_{i \in \mathcal{V}_{p} \setminus \{i\}} \sum_{m=1}^{M+1} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} + \sum_{j \in \mathcal{V} \setminus \{i\}} \sum_{m=1}^{M+1} \gamma_{\text{cat}}^{\text{cat}} h_{p, j}^{\text{cat}} x_{j, m, o}^{\text{cat}} \\
\quad \forall \text{cat} \in \{\text{laden, empty, repos}\}, \\
\begin{array}{c}
i \in \mathcal{V}, \\
t = 1, \ldots, T \\
p = 1, \ldots, P
\end{array}
\]  
(8)

Constraint (9) enforces the flow of loaded RTIs (orders) between origin and destination locations. It enforces the outbound flow of an origin node to be \( w_{p} \) and the inbound flow of a destination node to be \( -w_{p} \). Note that even though Constraints (9) are equality constraints, there is no obligation for an order to immediately be loaded and transported at its earliest pick up time. In a feasible solution, an order might be held for several time periods before loading. Similarly, the delivery date of an order is not definite.

\[
\begin{align*}
U_{p, t}^{\text{laden}} &= w_{p} \quad i = O(p), t = DT(p) \quad \forall i \in \mathcal{V}, \\
&= -w_{p} \quad i = D(p), t = DT(p) \quad \forall i \in \mathcal{V}, \\
&= 0 \quad \text{o.w.}
\end{align*}
\]  
(9)

\[
\begin{align*}
U_{p, t}^{\text{empty}} &= S_i \quad 0 < t = 0 \quad \forall i \in \mathcal{V}, \\
&= -S_i \quad i = O(p), t = DT(p) \quad \forall i \in \mathcal{V}, \\
&= 0 \quad \text{o.w.}
\end{align*}
\]  
(10)

\[
\begin{align*}
U_{p, t}^{\text{repos}} &= S_i \quad 0 < t = T \quad \forall i \in \mathcal{V}, \\
&= -S_i \quad i = D(p), t = DT(p) \quad \forall i \in \mathcal{V}, \\
&= 0 \quad \text{o.w.}
\end{align*}
\]  
(11)

Constraints (10) and (11), in a similar fashion enforce the flow of empty RTIs between origin and destination locations. The origin locations of the assigned empty RTIs are the RTI storage locations with \( S_{i} \). Their destination locations are the locations that they are needed to be loaded and transport the products (\( O(p) \), \( DT(p) \)). On the other hand, the repositioned empty RTIs need to get back to the storage locations. Therefore, the locations with \( S_{i} > 0 \) are their destinations and their origin locations are the locations that the loaded RTIs are unloaded (\( D(p), DT(p) \)).

Constraints (12) and (13) are logical constraints which are used to calculate the TTS of orders. Note that \( M \) is the classic “big \( M \)” based on these constraints then, if there is no flow of products on a specific arc \( (x = 0) \), that arc will not be included in the constraint on TTS \( (b = 0) \).

\[
\begin{align*}
\gamma_{\text{laden}}^{\text{laden}} h_{p, i}^{\text{laden}} x_{i, m, o}^{\text{laden}} &= \gamma_{p, i}^{\text{laden}} x_{i, m, o}^{\text{laden}} \quad \forall i \in \mathcal{V}, \\
p = 1, \ldots, P
\end{align*}
\]  
(12)

\[
\begin{align*}
\gamma_{\text{laden}}^{\text{laden}} h_{p, i}^{\text{laden}} x_{i, m, o}^{\text{laden}} &= M \gamma_{p, i}^{\text{laden}} x_{i, m, o}^{\text{laden}} \quad \forall i \in \mathcal{V}, \\
p = 1, \ldots, P
\end{align*}
\]  
(13)

Assuming \( F_{TTS} \) to be the temperature inside a vehicle of mode \( m \) and \( l_{i} \) to be the temperature inside a location \( i \) at time \( t \), based on Constraints (12) and (13), Constraint (14) states that for each order, the total time \( \times \) temperature of moving and handling an order must be less than or equal to the total required TTS of that order.

\[
\sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \leq F_{TTS} \quad \forall p = 1, \ldots, P
\]  
(14)

Let \( A_{m} = (a_{i, o}, m) \) be the set of all arcs of mode \( m \in \{1, \ldots, M\} \) crossing time period \( t \). Constraint (15) then states that in each time period, the number of used vehicles of a mode type must be less than or equal to a maximum value \( F_{V_{m}} \).

\[
\sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \leq F_{V_{m}} \quad \forall i \in \mathcal{V}, \\
m = 1, \ldots, M
\]  
(15)

### 4.4.2. Second stage

The objective function of the second stage is in the form of minimizing the ad-hoc costs related to predicted orders over scenarios \( \gamma_{1}, \ldots, \Gamma \). Terms (16a) represent flow costs of the loaded RTIs, the terms of (16b) are the flow costs of the assigned and repositioned empty RTIs respectively, the term (16c) shows locational costs of loaded RTIs, the terms (16d) represent locational costs of assigned, and repositioned RTIs, for each scenario orders. Finally, (16e) is the costs of using ad-hoc capacity increase in a scenario.

Similar to the first stage, the scenario-based repositioning decisions become positive, if it is required to return RTIs to their initial storage during the current planning horizon \( t = \psi_{p}, \psi_{p} + 1, \ldots, \psi_{p} + T \).

Constraints (17)–(20) are again variable pair matching of future scenario orders.

\[
\begin{align*}
\min \quad & \sum_{t=1}^{T} \sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \left[ \sum_{p=1}^{\Gamma} \gamma_{p, i}^{\text{laden}} x_{i, m, o}^{\text{laden}} \right] \\
+ \sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \left[ \sum_{p=1}^{\Gamma} \gamma_{p, i}^{\text{empty}} x_{i, m, o}^{\text{empty}} \right] \\
+ \sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \left[ \sum_{p=1}^{\Gamma} \gamma_{p, i}^{\text{repos}} x_{i, m, o}^{\text{repos}} \right] \\
+ \sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \left[ \sum_{p=1}^{\Gamma} \gamma_{p, i}^{\text{laden}} x_{i, m, o}^{\text{laden}} \right] \\
+ \sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \left[ \sum_{p=1}^{\Gamma} \gamma_{p, i}^{\text{empty}} x_{i, m, o}^{\text{empty}} \right] \\
+ \sum_{\text{cat}, m, o} \gamma_{\text{cat}}^{\text{cat}} h_{p, i}^{\text{cat}} x_{i, m, o}^{\text{cat}} \left[ \sum_{p=1}^{\Gamma} \gamma_{p, i}^{\text{repos}} x_{i, m, o}^{\text{repos}} \right]
\end{align*}
\]  
(16a–16e)
\[ γ_{p,i,j}^{\text{cat}}(m_1,m_2) = γ_{p,i,j}^{\text{cat}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

Constraints (23) and (24) enforce the
\[ γ_{p,i,j}^{\text{cat}}(m_1,m_2) = γ_{p,i,j}^{\text{cat}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

\[ b_{p,i,j}^{\text{alle}}(m_1,m_2) = b_{p,i,j}^{\text{alle}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

\[ b_{p,i,j}^{\text{alle}}(m_1,m_2) = b_{p,i,j}^{\text{alle}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

Constraints (21) are flow conservation constraints for future scenario orders.

\[ U_{p}^{\text{index}} = \sum_{p \in \{P-1\}} \sum_{m=1}^{M} \sum_{i,j=1}^{P} \gamma_{p,i,j}^{\text{cat}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, i \in \mathcal{V}, t = \phi, \phi + 1, \ldots, \phi + T, p' = 1, \ldots, P', \gamma = 1, \ldots, \Gamma \]

\[ U_{p}^{\text{index}} + φ_{p}^{\gamma} \]

assuming $S_{l_1}^{\text{rem}}$ in the second stage to be the \textit{remaining} inventory of empty RTIs after assigning some of them to the real orders in the first stage. Constraints (23) and (24) enforce the flow of empty RTIs (for known orders) between origin and destination locations. The origin locations of the assigned empty RTIs are the RTI storage locations with $S_{l_1}^{\text{rem}} > 0$. Their destination locations are the locations that are needed to be loaded and transport the products (Op,PT(p)).

If repositioning decisions are made in roll $\phi$, or in other words, $t_{\phi} \leq \phi + T$, the repositioned empty RTIs need to get back to their storage locations in the current roll. Therefore, the locations with $S_{l_1}^{\text{rem}} > 0$ are their destinations and their origin locations are the locations that the loaded RTIs are unloaded (Dp,DT(p)).

Constraints (25) and (26) are again logical constraints used in calculating the TTS of scenario orders in Constraints (27).

\[ x_{p,i,j}^{\text{lade}}(m_1,m_2) \geq \gamma_{p,i,j}^{\text{alle}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

\[ x_{p,i,j}^{\text{lade}}(m_1,m_2) \leq \gamma_{p,i,j}^{\text{alle}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

\[ \sum_{a_{i,j}(m_1,m_2)} p_{i,j}^{\text{alle}}(m_1,m_2) \leq \gamma_{p,i,j}^{\text{alle}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

\[ \forall a_{i,j}(m_1,m_2) \in \mathcal{A}, \phi = \phi, \phi + 1, \ldots, \phi + T, \gamma = 1, \ldots, \Gamma \]

Constraint (28) is the capacity constraint. In each scenario $\gamma$, the total number of RTIs (loaded or empty) for both real and predicted orders that is moved between $i$ and $j$ at time $t$, should be less than or equal to the total capacity of $y$ vehicles of mode $m$ transporting them, if that mode is chosen in that scenario.

\[ \sum_{p=1}^{P} \left( x_{p,i,j}^{\text{lade}}(m_1,m_2) + x_{p,i,j}^{\text{alle}}(m_1,m_2) \right) \leq \gamma_{p,i,j}^{\text{alle}}(m_1,m_2) \quad \forall p \in \{1, \ldots, P\}, \gamma = 1, \ldots, \Gamma \]

\[ \forall a_{i,j}(m_1,m_2) \in \mathcal{A}, \phi = \phi, \phi + 1, \ldots, \phi + T, \gamma = 1, \ldots, \Gamma \]

Finally, Constraints (29)–(33) define the nature of variables in the formulation.

This optimization problem with its roll-dependent right-hand side values and cost coefficients, is too complex to be solved by a state-of-the-art MIP solver, especially for real-sized problem instances. In the following section, we describe the ALNS algorithm used for the re-optimization process, and propose some updating strategies.
5. Adaptive Large Neighborhood Search

SteadieSeifi et al. (2017) presented an ALNS algorithm for the multi-modal transportation planning problem of perishable products and RTI repositioning. The algorithm is fast and in less than a computational time of 15 min, it provides good quality solutions for instances with around 2000 orders. In this section, we describe the adapted ALNS algorithm.

Fig. 3 shows our rolling horizon framework and the ALNS used for solving the STSP problem of roll \( \varphi \). Since this is a complete re-optimization process, we first destroy all parts of the roll solution \( z_\sigma \) that are not fixed or have not been executed yet. The destroyed parts include routes, repositioning orders, and some of the fleets that were not employed yet.

To anticipate future demand, we generate the set of our sample scenarios and add them to the set of existing and new orders.

The ALNS then generates an initial solution \( z_{\sigma+1} \) by first finding empty RTI reposition orders, and then, constructing routes for all laden and empty orders. For the destructed routes, it generates an initial plan for the destructed parts.

After generating the initial solution, we iteratively improve this solution by systematically using designated destroy and repair operators. The following destroy operators are used to remove the routes of orders:

- two inter-scenario random orders (R2r-A) operator,
- three inter-scenario random orders (R2r-A) operator,
- two inter-scenario similar orders (R2r-A) operator,
- three inter-scenario similar orders (R2r-B) operator,
- three intra-scenario similar orders (R2r-A) operator,
- three intra-scenario similar orders (R2r-B) operator,
- one ad-hoc positive order (Rd) operator.

The chosen orders can be laden, empty orders, or a combination of both. The number of possibilities in a random category is equal to the size of the order set and they are chosen randomly. On the other hand, the number of combinations in similar categories depends on the composition of the order set \( P \). For these operators, orders are chosen if they have similar origins or destinations, and if the difference in their pickup times or their delivery time is less than \( \beta \) time periods. \( \beta \) helps increasing the chances of consolidation.

The repair operators are as follows:

- cheapest path (Ip) operator,
- scheduling (Is) operator,
- consolidate two orders (Ic2) operator,
- consolidate three orders (single) (Ic3s) operator,
- consolidate three orders (paired) (Ic3p) operator,
- insert ad-hoc zero path (lipd) operator.

The cheapest path operator is similar to the Dijkstra classic shortest path algorithm, but instead of time, cost of the path is evaluated. The scheduling operator tries to find feasible consolidation options by putting RTIs on later schedules. For this purpose, two extra measure were created: the earliest departure time and the latest departure time of the modes. These feasible slack time for departing of each mode are recomputed and re-evaluated at each iteration of ALNS. Finally, the consolidation operators look for cheapest options by looking into all possible consolidated routes of the destroyed subnetwork and checking where their slack times overlap. In a single consolidation neighborhood Ic3s, the flows of the three orders are consolidated on one particular item in the fleet set, while in the paired consolidation neighborhood Ic3p, the flows of pairs of the orders are consolidated in two different items in the fleet set. The paired consolidation was designed to exploit for more complicated consolidation options which none of the other neighborhoods are capable of finding.

Compared to the ALNS of SteadieSeifi et al. (2017), with the redesigned set of operators, we give the algorithm better direction to work on both inter-scenario neighborhoods (A series), and intra-scenario neighborhoods (B series). In inter-scenario destroy neighborhoods, we only destroy paths of orders belonging to a particular scenario \( \gamma \) (or only the path of known orders) to let the algorithm to search neighborhoods improving a specific scenario. But in intra-scenario destroy neighborhoods, we look for improved fleet arrangement, satisfying the needs of more than one scenario (or combined with scenario \( \Gamma + 1 \)).

Moreover, the one ad-hoc positive order (Rd) destroy and insert ad-hoc zero path (lipd) repair operator are introduced to the stochastic ALNS in order to improve the neighborhood of orders \( P' \) with \( \theta_P > 0 \). The repair
operator Ipd tries to reinsert such orders into the solution so that its $\theta_{\varphi}$ becomes zero.

Besides redesigned destroy and repair operators, the proposed ALNS algorithm uses a Prioritization strategy to help a better search. Routes and plans of the current period have higher priority than the ones of later periods. To facilitate this, instead of randomly selecting orders, we define a time-based classification for choosing orders to destroy. Fig. 4 gives an illustration of this classification. The weights are $\alpha_1$, $\alpha_2$, $\alpha_3$. The more weight given to $\alpha_1$, the more concentration is on checking the earlier orders in the improvement of the roll solution.

As a stopping criterion, in this paper, we chose a CPU time limit for the improvement phase of ALNS. After ALNS is done, we execute the plans of roll $\varphi + 1$ and update the state of routes, the related orders, and the fleet. Execution of plans in practice can be translated as the instructions that are sent to relevant departments, to start loading/unloading operations, transportation, etc.

6. Computational results

In this section, we present the computational results, and test their sensitivity to different rolling horizon settings and policies.

Since this is a rolling horizon framework, it is not possible to directly compare our results to an exact equivalent, even for very small sizes of the problem. Therefore, we start by looking at the results of our proposed algorithm and the structure and cost elements of our multi-modal transportation of perishable products with empty RTI repositioning. Then, we explore the costs and lost customer orders for different RTI transportation modes with empty RTI repositioning. As a stopping criterion, in this paper, we chose a CPU time limit for the improvement phase of ALNS. After ALNS is done, we execute the plans of roll $\varphi + 1$ and update the state of routes, the related orders, and the fleet. Execution of plans in practice can be translated as the instructions that are sent to relevant departments, to start loading/unloading operations, transportation, etc.

6.1. Test sets and general parameters

The setup of our test sets and their inputs are similar to SteadieSeifi et al. (2017). The instances are categorized in three groups of 7, 11, and 20 hub locations, where the first group only includes the hubs in the Netherlands, the second group includes the hubs in the BeNeLux region, and the third group includes all hubs. We have three classes of truck, train, and barge transportation modes. We assume the truck connections to be available between all pairs of locations, but it is not the case for trains and barges. Table 1 gives the parameter setting for the modes where $f^m$ and $cap^m$ show the maximum number of available vehicles and their capacity for each mode type. $f^m$ is the given temperature for each mode type based on degree Celsius. $freq^m$ shows that after how many hours, the next vehicle of each mode type departs, which is used as their fixed schedules. $speed^m$ shows the average speed of each vehicle based on km/hr. Assuming Euro as the currency, $C^m_{\text{fix}}$, $C^m_{\text{laden}}$, and $C^m_{\text{empty}}$ are fixed costs of using a vehicle, variable costs of transporting full RTIs per hour, and variable costs of transporting empty RTIs per hour, respectively.

To create a real-time order arrival environment, we have built three instances $D = 1, 2, 3$, representing three large sets of hourly basis orders dispersed over the course of a month. We argue that the chosen instances represent enough variety to guarantee the generality of the analysis, especially due to their day-to-day and weekly variations.

Most of the settings and parameters are tuned similar to SteadieSeifi et al. (2017). Values for $\alpha_1$, $\alpha_2$, and $\alpha_3$ are set to 0.6, 0.2, and 0.1 respectively. The number of iterations where the best-found-solution is not improved, are set as $\tau = 1000$ and $\eta = 250$. Furthermore, a maximum computation time limit of 30 min is used in each roll. Note that in most of the analysis, $\tau$ is set to 12 (hours). In each instance $D = 1, 2, 3$, we will be (re-)planning 60 rolls.

6.2. General results

In this section, we solve our instances and show the distribution of total cost among fixed and variable costs, as well as multi-modal combinations. Table 2 shows that for our instances around 50% of the total costs are fixed costs paid for using the transport mode, and between 2% – 4% are transshipment costs. Moreover, the table shows that in all instances, less than 1% of demand is not fulfilled and is considered as lost sale.

We examined the number of used combination of modes in both forward and backward flows. For the forward flows, due to the critical quality requirements on products, the direct truck-only option is mostly used. For the backward flows, Table 3 shows more mode combination diversity in repositioning empty RTIs. Here, the interplay between transshipment and fixed costs influences the combinations: the more expensive transshipment costs are compared to fixed costs, the more direct transportation are witnessed. In contrast, cheaper transshipment increases the mode switch around the network.

Exploring these differences and trends in forward and backward

---

Table 1

<table>
<thead>
<tr>
<th>Transportation mode inputs.</th>
<th>$f^m$</th>
<th>$cap^m$</th>
<th>$f^m$</th>
<th>$freq^m$</th>
<th>$speed^m$</th>
<th>$C^m_{\text{fix}}$</th>
<th>$C^m_{\text{laden}}$</th>
<th>$C^m_{\text{empty}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truck services</td>
<td>200</td>
<td>22</td>
<td>5</td>
<td>1</td>
<td>65.00</td>
<td>136.34</td>
<td>6.412</td>
<td>2.849</td>
</tr>
<tr>
<td>Train services</td>
<td>100</td>
<td>1760</td>
<td>5</td>
<td>6</td>
<td>32.50</td>
<td>179.37</td>
<td>0.025</td>
<td>0.011</td>
</tr>
<tr>
<td>Barge services</td>
<td>100</td>
<td>704</td>
<td>5</td>
<td>2</td>
<td>18.52</td>
<td>118.04</td>
<td>0.008</td>
<td>0.004</td>
</tr>
</tbody>
</table>
flows, is not straightforward. There are many variables involved that could influence these behaviors. The time of the day that orders arrive, their delivery and TTS requirements, their volume and whether they can be consolidated with other orders of similar customer locations, are the first set of variables. Availability of multi-modal transportation for these locations, and their distance to inbound hubs is the second set of variables affecting the results.

### 6.3. RTI repositioning

Typically, a minimum inventory $S_i(\geq 0)$ of empty RTIs (e.g. 5000 trolleys) should be available at the inbound hub locations at beginning of each week. This is to enforce sufficient availability of the RTIs to manage the operations. In our rolling horizon framework, at the end of each day, we check the inventory level at these locations. If the end-of-the-week inventory level is less than $\xi = 100\%$ of $S_i$, empty RTI orders are generated to reposition RTIs.

In this section, smaller percentages for $\xi$ are tested to evaluate whether $\xi = 100\%$ inventory repositioning is necessary and at what costs. Table 4 compares the repositioning costs and lost sale of the rolling horizon solutions with 0, 20, 50, 80, and 100% inventory level percentages, compared to the target levels.

Table 4 shows that the combination of all new operators result in the best performance. Compared to the normal ANLS operators presented in SteadieSeif et al. (2017), the ratio of fixed costs but higher lost sales and more outsourced number of operators, and the intra-scenario operators, and the inter-scenario operators, and the inter-scenario positive operators yield higher performance in improving the solutions. Note that for this specific experiment, we switch off the ad-hoc operators (Rd) and (Ipd) to make sure the improvement percentages are not influenced by them. Table 5 shows that the combination of all new operators result in the best performance. Compared to the normal ANLS operators presented in SteadieSeif et al. (2017), the new operators guide the search better, while not increasing the computation time.

The ad-hoc operators (Rd) and (Ipd) are introduced to the ALNS algorithm to minimize the ad-hoc costs of outsourcing $\theta_p$ in scenarios. Table 6 compares the outsources volume, percentage of lost-sale in simulating the solutions, and the fixed costs for different operator combinations. The results show that without the operators (Rd) and (Ipd), we have less fixed costs but higher lost sales and more outsourced number of RTIs.

#### 6.4. Operator combinations

In this section, we discuss the efficiency of the proposed operators based on their share in improving the solution. For this purpose, we use the set $n = 20$, $D = 1$, and $r = 12$ (in total 60 rolls).

In section 5, we introduced three new categories of destroy operators: inter-scenario operators, intra-scenario operators, and the ad-hoc positive order (Rd) operator. For the first two categories, there are path, schedule, and consolidation based repair operators but for the third destroy operator, we have the insert ad-hoc zero path (Ipd) repair operator.

First, we compare the performance of our ALNS with different combinations of inter-scenario and intra-scenario operators. Since the number of all operator combinations is high, we defined three combinations. They give a nice overview on the contribution of the operators in improving the solutions. Note that for this specific experiment, we switch off the ad-hoc operators (Rd) and (Ipd) to make sure the improvement percentages are not influenced by them. Table 5 shows that the combination of all new operators result in the best performance. Compared to the normal ANLS operators presented in SteadieSeif et al. (2017), the new operators guide the search better, while not increasing the computation time.

The ad-hoc operators (Rd) and (Ipd) are introduced to the ALNS algorithm to minimize the ad-hoc costs of outsourcing $\theta_p$ in scenarios. Table 6 compares the outsources volume, percentage of lost-sale in simulating the solutions, and the fixed costs for different operator combinations. The results show that without the operators (Rd) and (Ipd), we have less fixed costs but higher lost sales and more outsourced number of RTIs.
and the minimum end-of-the-week repositioning rate of instance sets with 20 locations, 3 mode types of truck, train, and barge, for scenarios for future day demand are generated once per day (e.g. in the problem.

Observations are used to solve a formulation that is a cost function that is a function of the variables.

6.5. Scenario-based versus deterministic ALNS

In practice, instead of stochastic version of a problem, averages of past observations are used to solve a deterministic version of the operational problem.

Here, based on the daily demand distributions of customer locations, scenarios for future day demand are generated once per day (e.g. in the final roll of the day). We analyze the stochastic demand results on three instance sets with 20 locations, 3 mode types of truck, train, and barge, and the minimum end-of-the-week repositioning rate of ξ = 100%. The length of a roll r is equal to 24 h, which means that 90 rolls are run per instance.

Sample size Γ should be large enough to ensure solution reliability in any realization of the future. If this value is small, there is a risk that the number of scenarios does not represent the future adequately, and if it is too large, the computational burden for obtaining a good solution increases and might not be suitable for the rolling horizon framework.

Table 4
Different inventory repositioning percentages (the benchmark is ξ = 100%).

<table>
<thead>
<tr>
<th>n</th>
<th>n month</th>
<th>ξ = 100%</th>
<th>ξ = 80%</th>
<th>ξ = 50%</th>
<th>ξ = 20%</th>
<th>ξ = 0%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average Repositioning Cost</td>
<td>Avg. Lost Orders (%)</td>
<td>Repositioning Cost Diff. (%)</td>
<td>Avg. Lost Orders (%)</td>
<td>Repositioning Cost Diff. (%)</td>
<td>Avg. Lost Orders (%)</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>539,813</td>
<td>0.88</td>
<td>2</td>
<td>0.88</td>
<td>2</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>563,741</td>
<td>0.67</td>
<td>0</td>
<td>0.65</td>
<td>17</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>493,073</td>
<td>0.76</td>
<td>6</td>
<td>0.75</td>
<td>17</td>
</tr>
<tr>
<td>Average</td>
<td>–</td>
<td>0.77</td>
<td>–3</td>
<td>0.76</td>
<td>–12</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Table 5
Improvement percentage and computational time of different combination of operators.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Improvement</th>
<th>Total comp. time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old ALNS operators (SteadieSeifi et al. (2017))</td>
<td>11.54%</td>
<td>257</td>
</tr>
<tr>
<td>Only inter-scenario operators</td>
<td>2.03%</td>
<td>78</td>
</tr>
<tr>
<td>Only intra-scenario operators</td>
<td>13.57%</td>
<td>286</td>
</tr>
<tr>
<td>All new operators</td>
<td>16.79%</td>
<td>242</td>
</tr>
</tbody>
</table>

Table 6 also shows that the performance of the ALNS depends on the ad-hoc cost C_{0}. In this paper, the fixed costs C_{fix} range from 118 to 136 per vehicle. Table 6 shows that C_{0} higher than these result in θ_{D} to become zero at the expense of adding more vehicles to the fleet. Comparing the differences in the fleet composition however is not straightforward. There are various other factors impacting the schedules and the fleet.

Finally, Table 6 shows that the ad-hoc operators are not enough to find the best solution. To reduce lost sales and to minimize the fixed costs, using all the other operators are necessary.

Our ALNS algorithm keeps a short-term memory in each iteration, tracking down orders removed by each destroy operator. To diversify the search and avoid local optima, our ALNS algorithm also has a long-term memory of tracking chosen orders by the operators over all iterations.

Table 7 shows that this diversification step exploring all inter-scenario and intra-scenario improvement options results in the best performance.

Table 7
Impact of diversification.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Improvement</th>
<th>Total comp. time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No diversification</td>
<td>16.79%</td>
<td>242</td>
</tr>
<tr>
<td>Intra-scenario diversification</td>
<td>21.01%</td>
<td>312</td>
</tr>
<tr>
<td>All diversification</td>
<td>23.54%</td>
<td>355</td>
</tr>
</tbody>
</table>

Table 8 shows the performance of our stochastic model considering three sample sizes (Γ = 20, Γ = 50, and Γ = 100), compared to the deterministic one where daily averages are used as predictors of future orders.

The columns of Table 8 represent the costs of roll solutions for the deterministic model, the extra costs of simulating these deterministic solutions, the total deterministic and the stochastic solution costs. The table also shows the differences by percentage (Diff), as well as the percentage of rolls that stochastic solutions were better than their deterministic counterpart (Better rolls). The overall results show that on average, the stochastic model has lower average costs, showing the benefit of including demand uncertainty of demands into our model.

Increasing the sample size from 20 to 100, increases this benefit from an average of −4.20% to −12.24%, and the percentage of rolls with better solutions (following the stochastic model) increases from an average of 66%–80%. Therefore, to have in-sample stability, the sample size should be at least 100.

Fig. 5a-c shows the cost difference percentage of the instance with 20 locations for D = 3. These figure illustrate how the roll solution costs of stochastic problem become cheaper by increasing the sample size. However, they do not show similar behavior from one roll to another. Looking at these figures, days 2, 8, 14, and 24 show worse stochastic solutions than deterministic ones in all three cases. Days 2 and 8 are the beginning days of a week, day 14 is a weekend, and day 24 is a day in the middle of the week. It only can be concluded that increasing the sample size gives less days with more expensive stochastic solutions.

Table 6
Ad-hoc operators and resulting fleet costs.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Lost sale</th>
<th>Ad-hoc cap.</th>
<th>Total ad-hoc cost</th>
<th>Total Fixed cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without ad-hoc operator</td>
<td>1.07%</td>
<td>436</td>
<td>–</td>
<td>1,259,371</td>
</tr>
<tr>
<td>Only ad-hoc operator (C_{0} = 100)</td>
<td>0.92%</td>
<td>234</td>
<td>23,400</td>
<td>1,576,810</td>
</tr>
<tr>
<td>Only ad-hoc operator (C_{0} = 1,000)</td>
<td>0.89%</td>
<td>0</td>
<td>0</td>
<td>1,587,223</td>
</tr>
<tr>
<td>All operators (C_{0} = 1,000)</td>
<td>0.88%</td>
<td>0</td>
<td>0</td>
<td>1,404,635</td>
</tr>
</tbody>
</table>
Fig. 6a–c shows the difference between the number of vehicles that stochastic and deterministic solutions use in each mode type for each roll. When increasing the sample size, stochastic solutions tend to use less trucks and more trains in overall. The difference in barge use is not significant compared to the other two modes. Less trucks and more trains means cheaper fleet and more consolidation options.

6.6. The length of a roll

An important design aspect of our rolling horizon framework is the length of a roll. The longer $\tau$ is, the more data about new orders is collected and more orders are consolidated. Table 9 shows the results for roll lengths of 6, 12, 24, and 48 h.

By increasing $\tau$, the total system costs decrease by over 50%. This difference is even more pronounced for the fixed costs. The more accurate information is available on predicted orders, the more flexibility and consolidation options for transport. However, this comes at the price of increasing lost orders, which is the result of hard product quality requirements (especially for orders with far destinations among the 20 locations). By reducing $\tau$ from 12 to 6 h, the lost orders decrease, but the costs significantly increase, which shows that real-time decision making without further anticipation of future, results in solutions where the plans are not efficient and resources are wasted.

6.7. Partial versus complete re-optimization

In Section 5, we discussed that the periodically solved re-optimization problem inside our framework is a complete problem, where all unexecuted and unused elements of a roll solution are used to re-plan. In this section, we compare the results of a complete re-optimization framework to a partial re-optimization framework, where only new orders are considered in the re-optimization process.

Table 10 shows that without re-planning, fixed costs and total system costs increase by an average of 10.57% and 12.18%, respectively. Lost orders also increase without re-planning, but not as significant as the system costs. These results show that a complete re-optimization provides more consolidation opportunities by including all new and old orders into the re-planning.

7. Conclusions

In this paper, we studied the operational planning of multi-modal transportation of perishable products, taking product quality preservation and RTI management into account. We took the dynamics of order arrivals into account and incorporated uncertainty of demand into our operational planning.

We presented a rolling horizon framework where the MIP of SteadieSei et al. (2017) was transformed into a Scenario-based Two-stage Stochastic Program (STSP) with stochastic demand. This planning problem is periodically reoptimized by means of a modified version of the ALNS algorithm proposed in SteadieSei et al. (2017). At the first stage of the STSP, the fleet arrangement and all forward and backward RTI flows of realized orders are planned, and at the second stage, this arrangement and the remaining capacity and resources are used to provide optimal product flow and RTI repositioning, for a generated set of scenarios.

To show the performance of our proposed solution algorithm, we analyzed the sensitivity of our results to various factors, from behaviour of demand and the sample size, to the roll length and re-optimization strategies. The results showed the trade-off between the computational effort and the solution costs. In addition, in comparison to an equivalent deterministic algorithm, our results showed the cost savings of including demand volatility in operational planning.

A possible future work can be to include other types of uncertainty sources into the planning, such as network disruption sources, i.e. transport capacity cancellations, accidents, and delays. Another future research possibility is to find strategies to handle multi-roll scenario samples to explore further consolidation possibilities. We tested our framework on the horticultural industry of the Netherlands with its demand trends and a strict TTS limit. A future research possibility is to test our solution algorithm on different perishable produce industries with different customer demand and TTS limits.
Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The study in this article has been supported by the DaVinc3i project with the reference number 2010.2.034R.
References


