A metaheuristic for the multimodal network flow problem with product quality preservation and empty repositioning

Transportation Research Part B 106 (2017) 321–344

Contents lists available at ScienceDirect

Transportation Research Part B

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

A metaheuristic for the multimodal network flow problem with product quality preservation and empty repositioning

M. SteadieSeifi*, N.P. Dellaert, W. Nuijten, T. Van Woensel

School of Industrial Engineering and Innovation Sciences, Technical University of Eindhoven, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands

ARTICLE INFO

Article history:
Received 27 June 2016
Revised 26 June 2017
Accepted 18 July 2017
Available online 2 August 2017

Keywords:
Multimodal transportation
Mixed Integer Programming (MIP)
Adaptive Large Neighborhood Search (ANLS)
Reusable transport item
Perishability

ABSTRACT

We study a transportation planning problem with multiple transportation modes, perishable products, and management of Reusable Transport Items (RTIs). This problem is inspired by the European horticultural chain. We present a Mixed Integer Programming (MIP) optimization model which is an extension of the Fixed-charge Capacitated Multi-commodity Network Flow Problem (FCMNFP). The MIP integrates dynamic allocation, flow, and repositioning of the RTIs in order to find the trade-off between product freshness requirements, and operational circumstances and costs. We furthermore propose an Adaptive Large Neighborhood Search (ALNS) algorithm with new neighborhoods, and intensification and diversification strategies. We then provide detailed computational analysis on its properties, compare its results with a state-of-the-art MIP solver, and provide practical insights.

1. Introduction

Perishable supply chains are international businesses and the horticultural industry of the Netherlands is an example. Everyday, around 400,000 types of cut flowers and plants, with an average daily turnover of more than 8 million Euros (FloraHolland, 2015), are transported from around the world to the auction houses in the Netherlands, to get auctioned, sold, and further transported throughout Europe and beyond. Kenya, Ethiopia, Israel, Belgium, Germany are among the top producers of these products, and United Kingdom, Netherlands, Germany, France, Italy, Poland, and Russia are among the top European markets.

These perishable products are fragile, have short shelf-lives, and travel long and in different climates. Temperature fluctuation and long handling time have a direct influence on their deterioration. To ensure freshness of the products, they are traditionally transported by either air or road, and in temperature-controlled environment. However, their market is growing, and adding more air and road vehicles results in more expensive transportation, and causes various social and environmental issues such as congestion and pollution. The horticultural supply chain of the Netherlands is aiming to lower related logistics costs in 2020 by 15%, equivalent to 64 million Euros (FloraHolland, 2015).

The challenge of finding the optimal transportation fleet plan is added to other operational issues such as resource management. An optimal transportation of perishable products needs synchronized flow with minimum waiting and handling. In the horticultural supply chain, this is highly dependent on the availability of Reusable Transport Items (RTIs). An RTI is an empty loading unit which can have different sizes ranging from a small box to a large 45-feet container. Their number

* Corresponding author.
E-mail address: maryam.steady@gmail.com (M. SteadieSeifi).

http://dx.doi.org/10.1016/j.trb.2017.07.007
0191-2615/

is limited, and their shortage results in quality decay of the products awaiting them, and therefore, in less profit. Returning or repositioning these units is costly and does not bring any direct profit. As a result, solutions integrating the forward flow of loaded RTIs with the backward flow of empty ones are needed to minimize the system-wide costs.

A cheap, diverse, flexible, and environmentally friendly transportation, ensuring freshness of the products while offering a competitive price, requires consolidation and switching from air and road to other modes of transport. *Multimodal transportation* has gained a lot of attention over the last 50 years (SteadieSeifi et al., 2014). Other industries are not unfamiliar with the opportunities that it offers. In perishable supply chains though, switching to slower modes means an increase in transportation time, which furthermore might decrease the product quality. Finding the trade-off between minimizing operational costs and product quality preservation becomes an interesting research subject, which is the target of our research.

Our contributions are as follows. In this paper, we study the long-haul transportation of perishable products. In order to include product preservation requirements, we present a mode-space-time network where all types of multimodal operations such as holding, handling, transshipment, and transportation are included. We model the problem as a Mixed-Integer Program (MIP), where we add new sets of constraints to the classic Fixed-charge Capacitated Multicommodity Network Flow Problem (FCMNFP). These constraints include a product quality measure based on temperature and travel time, and enforces a maximum limit on the products after which the products are spoiled. Moreover, we integrate the forward flow of loaded RTIs with the backward flow of empty ones via a set of special constraints. Based on a given demand, these constraints automatically assign and move the needed empty RTIs. Balakrishnan et al. (1997) have shown that the capacitated fixed-charge network design problem is NP-hard. Our problem incorporates additional resource management constraints which add further complexity to the problem. The number of decision variables goes very quickly beyond what can be solved to optimality with a state-of-the-art MIP solver. Therefore, in this paper, we build upon the literature and propose an ALNS algorithm with new operators, improved scoring mechanism, and extra strategies, to solve this problem.

In the remainder of this paper, Section 2 gives an overview on the literature of transportation of perishable products, as well as asset management issues in transportation, and the related solution algorithms. Section 3 describes the problem, and presents its MIP formulation. In Section 4, the proposed solution algorithm is explained and in Section 5, its performance is analyzed and compared with a state-of-the-art solver. Finally, in Section 6, we give some concluding remarks and describe potential future work.

2. Literature review

Planning transportation of perishable products in the literature is identified by extra preservation constraints or penalty costs. We can group transportation problems into long-haul transportation, and last-mile (or first-mile) transportation.

Last-mile (or first-mile) distribution problems are traditionally modeled as Vehicle Routing Problems with Time Windows (VRPTW) and the goal is to find the optimal load, delivery routes, and departure times of the fleet of vehicles. Doerner et al. (2008) study the Pickup and Delivery Problem (PDP) of blood products with strict time windows. Hsu et al. (2007) model a food distribution planning problem with stochastic and time-dependent travel times and time-varying temperature. Osvald and Stirn (2008) also address distribution of fresh vegetables with time-dependent travel times, but add a quality degradation based cost function to the objective function. Tarantilis and Kiranoudis (2001, 2002) study the distribution of fresh milk with a heterogeneous fixed fleet, and the distribution of fresh meat in a multi-depot network, respectively. They consider strict time windows for delivery of the products. Derigs et al. (2011) and Mendoza et al. (2011) study distribution planning problems for food and petrol where the products are incompatible and they should be separated and allocated to separate trucks, while customers order them simultaneously, resulting in excessive transportation costs.

On long-haul transportation of perishable products, Reis and Leal (2015) propose a MIP model for a soybean shipping chain planning problem where choice of transportation mode is included in the model besides decisions for annual crop purchase. Since their real-world application deals with significant uncertainty related to crop production, they define several combinations of scenarios for this uncertainty and apply their MIP model to each scenario in order to give insights for their decision makers. Studying long-haul transportation of perishable products is slowly getting more attention.

In this paper, we add to this literature of long-haul transportation of perishable products by incorporating management of resources, here RTIs, into our multimodal planning problem, which is missing in Reis and Leal (2015). An optimal transportation comprises the optimal and timely utilization and operation of its resources, called assets. *Assets* can be RTIs, vehicles, crews, power units, engines, etc (SteadieSeifi et al., 2014). Positioning, balancing, allocating, repositioning, and rotation of assets are the subject of asset management.

Repositioning of vehicles has been widely investigated in the literature, mostly addressed as Service Network Design (SND) problems. SND problems with cyclic service design account for returning of empty vehicles to their service starting location. Pedersen et al. (2009), Andersen et al. (2009a, 2009b, 2011) present comprehensive studies on formulating SND problems with vehicle repositioning and their computational differences. Moccia et al. (2011) propose a column generation heuristic for a rail and road transportation system with both consolidated and dedicated services. Thiongane et al. (2015) introduce new constraints to the fixed-charge capacitated multicommodity network design problem, where commodity volume is not splittable, and there is a limit on the number of arcs that each commodity can travel between its origin and destination nodes. They propose different formulations and several relaxation methods to solve them. Li et al. (2016) formulate a SND problem with heterogeneous vehicles, but decompose it into two interdependent sub-problems of a fixed-charge capacitated multicommodity network design problem, and a vehicle assignment problem (VAP). Unlike most of the SND

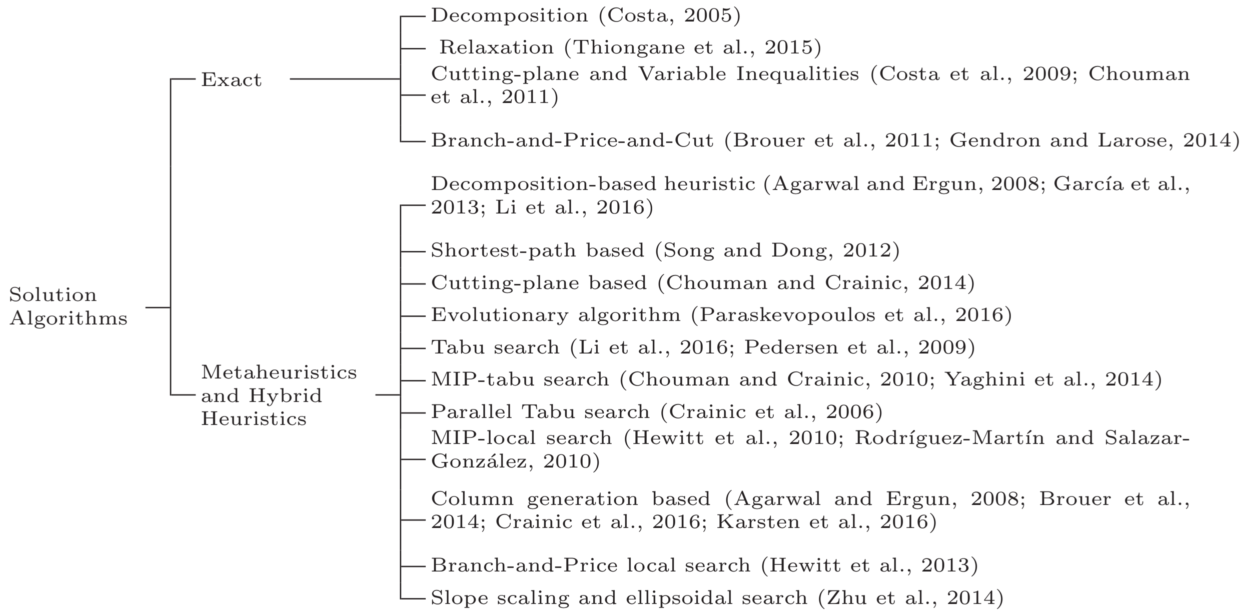


Fig. 1. OR literature on solving Fixed-cost Capacitated Multi-commodity Network Flow Problem. Rodríguez-Martín and Salazar-González (2010); Paraskevopoulos et al. (2016); Karsten et al. (2016); Hewitt et al. (2010, 2013); Gendron and Larose (2014); García et al. (2013); Crainic et al. (2006, 2016); Brouer et al. (2014); Agarwal and Ergun (2008).

literature where planning horizons are discretized, Demir et al. (2015) model their multimodal SND with stochastic travel times and emission costs as a continuous-time mixed-integer linear program, and apply a Sample Average Approximation (SAA) algorithm to find the most robust plan.

The number of studies managing loading units such as RTIs is comparably limited. Choong et al. (2002) model the flow of owned and leased empty containers on barges, and investigate 15-day and 30-day planning horizons. Choong et al. (2002) however do not study the integration of empty repositioning and forward flow of products. Braekers et al. (2013) and Meng and Wang (2011) are two examples we found where empty container repositioning is included into liner shipping service design. However, in both papers, the repositioning orders are given and pre-defined, and can be treated as additional orders besides the usual product orders. Zhu et al. (2014) is an example of simultaneous planning of multiple type of assets in SND problems. They study a rail system with car classification, car blocking, and train makeup, and model it as a three-layered network (for service, block, and car). They solve this problem by means of a hybrid metaheuristic algorithm combining slope scaling, enhanced by long-term memory-based perturbation strategies and an ellipsoidal search method. Erera et al. (2005), Brouer et al. (2011), Song and Dong (2012) are the examples of the research in integrating forward commodity flow with backward empty repositioning flow into a model. Erera et al. (2005) propose a multicommodity network flow formulation, and compare a base repositioning strategy (a current state-of-the-practice) with three alternative strategies which integrate the repositioning and routing of the containers simultaneously. These three strategies are weekly, bounded daily, and unbounded daily repositioning. Brouer et al. (2011) model a liner shipping network flow problem with repositioning and propose arc-flow and path-flow formulations for it. Song and Dong (2012) study a joint cargo routing and empty container repositioning in a shipping network, propose a continuous inventory model in order to minimize the total relevant costs including all terminal operational and demurrage costs, customer demand backlog costs, and empty container inventory costs at ports. However, these papers do not include all dynamics of multimodal operations into their network modeling.

Our long-haul transportation planning problem with management of RTIs is a special case of FCMNFP problems. Fig. 1 gives an overview of different approaches in the literature to solve FCMNFP problems.

Among the research works on exact solution algorithms, Costa et al. (2009) have studied various inequalities, such as Benders, metric, and cutset inequalities. Chouman et al. (2011) also present a cutting plane algorithm where it implements the strong cover, minimum cardinality, flow cover, and flow pack inequalities. They compare and discuss the strength of each of these inequalities. On the other hand, Costa (2005) provide a review on the Benders decomposition techniques in different network design problems. Chouman and Crainic (2014) combine a cutting-plane procedure with a variable-fixing procedure to solve their SND problem, embedding a learning mechanism to the cutting-plane procedure to identify promising variables. Thiongane et al. (2015) apply several relaxation methods, such as Lagrangian relaxation, linear programming relaxation, and partial relaxations of the integrality constraints, and conclude that Lagrangian relaxation seems to offer a good tradeoff between the bound quality and the computation time.

Among metaheuristic algorithms, advanced Tabu Search (TS) algorithms have been proposed the most. Pedersen et al. (2009) present a two-stage TS algorithm for the arc-based Design-Balanced Capacitated Multicommodity Network Design

(DBCMND) problem. The first stage of their TS explores the design-defined solution space via single-path-based neighborhoods. Then in the second stage, feasible solutions are identified. Chouman and Crainic (2010) integrate mathematical programming with a TS algorithm for the FCMNFP problems. In their algorithm, the MIP with a cutting-plane method computes a lower bound, while the TS explores the solution space for upper bounds. They also propose a new arc-balanced cycle-based neighborhood. Yaghini et al. (2014) also introduce a new cutting-plane neighborhood into their TS metaheuristic, which uses different strategies to select an open arc to be closed. They show that their neighborhood outperforms many other methods such as cycle-based, path relinking, multilevel, and local branching methods.

Looking at all these research works, we see that they are not able to solve problems with more than 100,000 variables, while in our problem, we need to handle more variables and a similar number of constraints. Therefore, we need an algorithm which can search the solution space fast but in a smart way. Large Neighborhood Search (LNS) algorithms, and in particular, Adaptive LNS (ALNS), have shown great performance for various large-scale transportation problems. ALNS was first introduced by Ropke and Pisinger (2006). They propose ALNS to solve a pickup and delivery problem with time windows. Pisinger and Ropke (2007) later propose a more general ALNS to solve various problems including the Vehicle Routing Problem with Time Windows (VRPTW) and the Capacitated Vehicle Routing Problem (CVRP). ALNS has been widely applied to VRP problems, but it is also gaining the attention of researchers in other areas. Examples are Grangier et al. (2016) for solving a two-echelon VRP with multi-trips at the second echelon, Gharehgozli et al. (2014) for scheduling twin automated cranes in a yard, Hemmati et al. (2016) for solving the inventory-routing with heterogeneous fleet of ships, Mauri et al. (2016) for a berth allocation problem, Rifai et al. (2016) for the distributed permutation flow shop scheduling problem with multiple factories, and Canca et al. (2017) for solving a nonlinear location-routing problem, to design of rapid rail transit network. ALNS is relatively new to the literature, and its capabilities and limitations are still under investigation. It has not been applied to FCMNFP problems, therefore in this paper, we add to the literature and propose an ALNS algorithm to solve the long-haul transportation planning of perishable products and RTI repositioning.

In the next two sections, we define the problem and explain its mathematical properties.

3. Problem description and mathematical model

In this section, we provide the problem description, the definitions, and the assumptions, and explain how we model it. Later, we present its mathematical formulation.

3.1. Problem description

The long-haul transportation of perishable products with repositioning of empty RTIs is an extension of the Fixed-charge Capacitated Multicommodity Network Flow Problem (FCMNFP), 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 throughout the network is limited and their flow is subjected to strict resource balance constraints. For instance, the number of RTIs available at the beginning of the planning horizon should be equal to their number at the end of horizon. Note that the initial number of RTIs is given.

There are two reasons making RTI management different from managing the fleet of vehicles that has been widely studied in the literature. First, the number of RTIs around the network is usually much larger than the number of vehicles. Consequently, indexing an RTI in the modeling like how it is done for vehicles, would result in an enormous intractable model. Secondly, if there are more than one storage center for the empty RTIs, it is usually not important to return an RTI to the same location where it was loaded. Usually, vehicles need to return to the same location they started their travel, while the only requirement for RTI management is, for instance, to have as many RTIs at the end of a planning horizon as there was at the beginning (or in other words, RTIs are anonymous). These reasons make RTI management unique.

Demand is here represented by orders. An 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 shows the number of RTIs (e.g. 2 trolleys) needed to transport products from the origin to the destination. The 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. These schedules are not definite though, and as long as the order is transported within the length of this schedule, its flow is feasible. For instance, an order might be held for a few hours at its origin destination before it is loaded and transported.

Modeling the product freshness, evaluating its shelf life, and translating it into the market value of the product, are not straightforward. There is no standard model showing how the product quality changes over the course of its transportation. There are various physical, chemical, and biological factors involved. In practice, quality is translated as how much customers are willing to pay for it. Customers' desired freshness (e.g. how much a rose bud is open, or how mature a cheese product is) depends of the type of product, the market it is sold to, seasonality, etc. *Shelf life* of a product (or vase life of a flower) is the length of time after which products are perished and have no value. In modeling the product quality, if product value is constant over its shelf life, e.g. for blood (Doerner et al., 2008), quality preservation is modeled as a constraint (e.g. enforcing a time threshold). Otherwise, a quality loss function is defined and relevant penalties for quality decrease are added to the main objective function of the planning problem (Ahumada and Villalobos, 2011; Hsu et al., 2007). The literature hardly includes temperature conditions in the quality control modeling. In horticultural industry of the Netherlands, product freshness is approximated by a Time Temperature Sum (TTS) measure. TTS is a simple linear measure used in some industries representing the total time that products can be transported in different temperature regimes (Tromp et al., 2012; Sloof and Everest, 2001; Taoukis, 2001). For instance, if a loaded RTI is moved on a train for 5 hours and the temperature inside the train is 10 °C, the TTS of this travel is 50. Tromp et al. (2012) show that the Time Temperature Sum is a good predictor of the quality of cut roses, especially in the range of 2–6 °C. For higher temperatures, the TTS seems be underestimating the quality, but since transportation of flowers is usually in the temperature range of 2 – 6°C, this predictor can be used without much loss of applicability. In this paper, a maximum allowed TTS (e.g. 200 hour-degree) is enforced for each order, and this limitation is added to the model via a new set of constraints.

Typically, demand in a FCMNFP problem is defined as commodities, and the only flow decision is defined on their routing throughout the network. However, in this problem, there are three types of flow decisions: 1) *laden* flow decisions for loaded RTIs that are transporting the products, 2) *assign* flow decisions for empty RTIs to move from a location with surplus of RTIs, to the origin location of an order to be assigned to, and 3) *repos* flow decisions for empty RTIs to be repositioned from the destination locations of orders back to storage locations with RTI shortage.

In order to transport these RTIs, we have a fleets of different transportation modes available throughout the network. Each transport mode has its own schedules, which we assume to be given, and its vehicles are operated based on these schedules. We assume a specific temperature for each mode which for example shows the temperature inside a truck trailer, a train car, or a barge storage room. Of course, the vehicles of a transport mode are capacitated (based on RTIs), and there are limited number of vehicles available for each transport modes.

Besides transportation of orders, their related handling or holding activities also have an influence on their quality decay. Therefore, the hub locations where orders are handled or held, also have specific temperature which is included in the TTS calculations. In contrast, locations are uncapacitated regarding the number of operations they can do, or the number of vehicles they can accommodate.

The objective of this problem is to minimize total system costs, which can consist of for instance, costs of renting a vehicle or a service, reserving space (capacity) on a freight train, costs of handling (loading, unloading, or transshipping) products or RTIs, costs of storing them, etc., and different penalties for failing to provide the customers' requested service level or product freshness. Basically, any cost, enforced by operating companies such as carriers, forwarders, shippers, and Logistics Service Providers (LSPs), can be a part of the objective function. An objective function can have various forms, such as maximizing revenue, customer service levels, or even maximizing product freshness, but in this paper, operational costs are used as the performance indicators.

3.2. Mode-space-time network

The physical transportation network is characterized by nodes $i \in \mathcal{N}$ representing the hub locations, and the arcs (*i*, *j*) representing different routes connecting these locations. Between each location pair, at least one transportation mode $m \in \{1, \ldots, \mathcal{M}\}$ can operate. Truck, train, and barge are examples of the possible modes.

In order to include all activities such as handling and holding operations into the model, we transform the physical network into a *mode-space-time representation*. First, we divide the time horizon T (e.g. 48 hours) into a set of time periods $t = 1, \ldots, T$ (e.g. an hour), and map the physical network in both time and space. Each node in such a network represents a location at a time period. Of course, time has a continuous nature, and discretizing it is not the only modeling approach. In continuous-time models, modeling the precedence of loading, unloading, transporting, and other activities, and their scheduling synchronization is difficult, especially in the context of multimodal transportation, where some modes such as trains and barges have their own timetables. In comparison, in discrete-time models, if t is really small compared to T (e.g. t has a length of one minute and T is a month), the size of the network will be too large to handle for any MIP solver. However, in long-haul transportation, length of activities are usually aggregated to e.g. hourly scales, and the time variable in the product freshness measure of our problem is not continuous. Therefore, in this research, we choose to discretize the time dimension.

The space-time mapping is furthermore copied $\mathcal{M}+1$ times, and therefore, each node $v \in \mathcal{V}$ in this new network represents a location $i \in \mathcal{N}$ at a time $t \in \{1, \ldots, T\}$ period on a mode $m \in \{1, \ldots, \mathcal{M}+1\}$. Layers $m = 1, \ldots, \mathcal{M}$ can accommodate all transport activities, whether with fixed timetables or flexible, but the extra layer of $m = \mathcal{M}+1$, here called *holding mode H*, is added to enable modeling handling and holding activities. Fig. 2 gives an illustration of a simple mode-space-time network with 3 locations, 16 time periods, and 2 modes. In this example, a feasible flow itinerary for an order from location 1 to location 3 with PT of 5 and DT of 9 is shown. All empty RTIs are kept at location 2, and at the end of planning horizon, their total number should be equal to their initial number. For the order, the needed RTIs are first loaded to vehicles of mode 2, transported from location 2 to location 1, and unloaded at location 1 at time 4. Then the RTIs are filled and loaded to vehicles of mode 1, moved to location 3, and unloaded at location 3 at time 7. After unloading and emptying the RTIs, they are returned to their storage location via mode 2 in a similar fashion, and are kept for the final 2 consecutive periods.

The mode-space-time network has the flexibility to accommodate any type of multimodal structure. For example, one mode can have a complete network, which is reasonable for trucks, or have a scheduled incomplete network, such as rail-

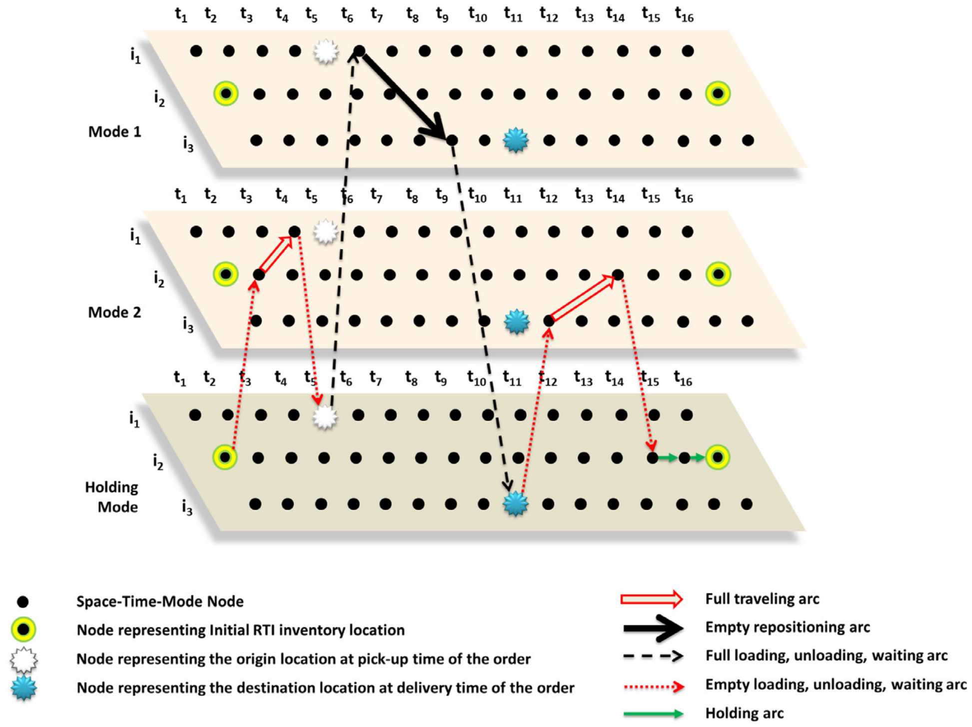
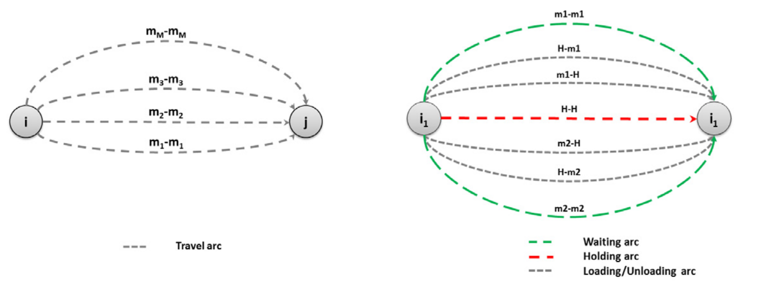


Fig. 2. An example of the mode-space-time representation of a flow network problem.



(a) Travel arcs connecting different nodes (b) Location arcs representing location operations

Fig. 3. An illustration of different arcs in the model.

ways and waterways where trains and barges are not available for all pairs of locations, and have their own departure timetables.

Now, in this mode-space-time network, a feasible arc $a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V} \times \mathcal{V})$, represents four types of operations:

(i) a *travel arc* for traveling between hub locations (i, j), leaving at particular time t by a particular mode type $m_1 = m_2 = m$ (Fig. 3a). Depending on its departure time t, a travel arc has a length of $r^{(m)}_{(i,j),t}$.

(ii) a *loading/unloading arc* for loading RTIs to a particular mode m (or unloading from it) at a location $i = j$. A loading arc has a modal state of $(m_1, m_2) = (H, m)$ and an unloading arc has a modal state of $(m_1, m_2) = (m, H)$, and like travel arcs, it can have different length of $r^{(m_1,m_2)}_{i,t}$ depending on the location and time t.

(iii) a *waiting arc* representing the stand-by state of a mode $m_1 = m_2 = m$ (e.g. for switching rail tracks at borders, or custom clearance) at a location $i = j$ at time t. The length of waiting arc is one.

(iv) a *holding arc* for holding RTIs at a location $i = j$ at time t for one time period.

Since the last three activities occur at hub locations, the arcs representing them are called *location arcs* (Fig. 3b).

Let $A_1 = \{a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V} \times \mathcal{V}) \mid i \neq j, m_1 = m_2\}$ be the set of all feasible and given *travel arcs* in the mode-space-time network, and let $A_2 = \{a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V} \times \mathcal{V}) \mid i = j, m_1 = H \parallel m_2 = H\}$ be the set of all feasible *location arcs* in the network. Similarly, let $A_m = \{a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V} \times \mathcal{V}) \mid m_1 = H \parallel m_2 = H\}$ be the set of all feasible loading, unloading, traveling, and waiting arcs in the network related to mode m. In the remainder of this section, we introduce our MIP.

3.3. Mathematical model

We present the mathematical formulation for our multimodal network flow problem with product quality preservation and management of RTIs.

An order p is characterized by its origin $O(p)$, its destination $D(p)$, volume w_p, an earliest pick up time of $PT(p)$, a latest delivery due date of $DT(p)$, and the maximum allowed TTS of $\mathcal{L}_p$.

Each transport mode $m \in \{1, \ldots, \mathcal{M}\}$ has its given schedules, its speed $speed^m$, vehicle capacity cap^m, total F^m number of available vehicles, and a temperature l^m. In our problem, costs associated with each transport mode are the fixed costs of using a vehicle (or its available capacity) C^m, and the variable costs of C^m_{laden} per loaded RTI per time period, and C^m_{empty} per empty RTI per time period.

Each location i is assumed to have a temperature $l_{i,\,t}$ at time t, and a unit holding cost of C_i^{hold} per RTI. Moreover, we also assume each location i to have $S_i \geq 0$ number of RTIs available at the beginning of the planning horizon, and we want the number of RTIs at the end of horizon to be equal to the initial value of S_i. Let $C_i^{(m_1,m_2)}$ be the general term for loading, unloading, and holding costs per RTI per time period, which are enforced on the location arcs.

As described earlier, for each order p, there are three flow decisions in our problem: *laden, assign*, and *repos* flow decisions. In our mode-space-time network, for each flow decision, we will have two sets of decision variables: 1) $\check{x}$ to show how many RTIs *enter* an arc $a_{(i,j),t,(m_1,m_2)}$, and 2) $\hat{x}$ to show how many RTIs *exit* the arc. It would have been easier to model, if we had added an extra index t' to the (i, j), t, (m_1, m_2) indices, in oder to distinguish the starting and the finishing times of a travel or an operation, and define only one x variable set. However, our model has in total $T/2$ times less variables, which reduces the size of the model.

Similarly, there are two sets of variables $\check{y}$ and $\hat{y}$ to show the number of vehicles respectively entering and exiting an arc. The auxiliary binary variables $\check{b}$ and $\hat{b}$ are also added to help calculating TTS.

Three categories of variables U are 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 FCMNFP model with product quality preservation and empty RTI management is then formulated as follows:

$$\min$$

$$\sum_{a_{(i,j),t,(m_1,m_2)} \in A_1} r^{(m)}_{(i,j),t} \left[\sum_{p=1}^{\mathcal{P}} C^m_{laden} \left(\check{x}^{\text{laden}}_{p,(i,j),t,(m_1,m_2)} \right) \right] \tag{1a}$$

$$+ \sum_{a_{(i,j),t,(m_1,m_2)} \in A_1} r^{(m)}_{(i,j),t} \left[\sum_{p=1}^{\mathcal{P}} C^m_{empty} \left(\check{x}^{\text{assign}}_{p,(i,j),t,(m_1,m_2)} + \check{x}^{\text{repos}}_{p,(i,j),t,(m_1,m_2)} \right) \right] \tag{1b}$$

$$+ \sum_{a_{(i,j),t,(m_1,m_2)} \in A_2} r^{(m_1,m_2)}_{i,t} \left[\sum_{p=1}^{\mathcal{P}} C_i^{(m_1,m_2)} \left(\check{x}^{\text{laden}}_{p,(i,i),t,(m_1,m_2)} \right) \right] \tag{1c}$$

$$+ \sum_{a_{(i,j),t,(m_1,m_2)} \in A_2} r^{(m_1,m_2)}_{i,t} \left[\sum_{p=1}^{\mathcal{P}} C_i^{(m_1,m_2)} \left(\check{x}^{\text{assign}}_{p,(i,i),t,(m_1,m_2)} + \check{x}^{\text{repos}}_{p,(i,i),t,(m_1,m_2)} \right) \right] \tag{1d}$$

$$+ \sum_{m=1}^{\mathcal{M}} \sum_{a_{(i,j),t,(m_1,m_2)} \in A_m} C^m \times \check{y}_{(i,j),t,(m_1,m_2)} \tag{1e}$$

The objective function is in the form of minimizing total system costs, where the term (1a) represent flow costs of the loaded RTIs, the terms of (1b) are the flow costs of the assigned and repositioned empty RTIs respectively, the term (1c) shows locational costs of loaded RTIs, the terms (1d) represent locational costs of assigned, and repositioned RTIs, and finally, (1e) shows the costs of using the modes.

Constraints (2)–(7) matches pairs of entering and exiting variables, to ensure network arc connectivity.

$$\check{x}^{\text{cat}}_{p,(i,j),t,(m_1,m_2)} = \hat{x}^{\text{cat}}_{p,(i,j),(t+r^m_{(i,j),t}),(m_1,m_2)} \qquad \forall cat \in \{laden, empty, repos\},\ a_{(i,j),t,(m_1,m_2)} \in A_1,\ p = 1, \ldots \mathcal{P} \tag{2}$$

$$\check{x}^{\text{cat}}_{p,(i,i),t,(m_1,m_2)} = \hat{x}^{\text{cat}}_{p,(i,j),(t+r^{(m_1,m_2)}_{i,t}),(m_1,m_2)} \qquad \forall cat \in \{laden, empty, repos\},\ a_{(i,j),t,(m_1,m_2)} \in A_2,\ p = 1, \ldots \mathcal{P} \tag{3}$$

$$\check{y}_{(i,j),t,(m_1,m_2)} = \hat{y}_{(i,j),(t+r^m_{(i,j),t}),(m_1,m_2)} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A_1 \tag{4}$$

$$\check{y}_{(i,i),t,(m_1,m_2)} = \hat{y}_{(i,j),(t+r^{(m_1,m_2)}_{i,t}),(m_1,m_2)} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A_2 \tag{5}$$

$$\check{b}_{p,(i,j),t,(m_1,m_2)} = \hat{b}_{p,(i,j),(t+r^m_{(i,j),t}),(m_1,m_2)} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A_1,\ p = 1,\dots\mathcal{P} \tag{6}$$

$$\check{b}_{p,(i,i),t,(m_1,m_2)} = \hat{b}_{p,(i,j),(t+r^{(m_1,m_2)}_{i,t}),(m_1,m_2)} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A_2,\ p = 1,\dots\mathcal{P} \tag{7}$$

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.

$$U^{cat}_{pit} = \sum_{j\in\mathcal{V}-\{i\}} \sum_{t'>t} \sum_{m=1}^{\mathcal{M}+1} \check{x}^{cat}_{p,(i,j),t,(m_1,m_2)} - \sum_{j\in\mathcal{V}-\{i\}} \sum_{t'<t} \sum_{m=1}^{\mathcal{M}+1} \hat{x}^{cat}_{p,(j,i),t,(m_1,m_2)} \qquad \forall cat \in \{laden, empty, repos\},\ i \in \mathcal{V},\ t = 1,\dots,T,\ p = 1,\dots\mathcal{P} \tag{8}$$

Constraint set (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 due date of an order is not definite.

$$U^{laden}_{pit} \begin{cases} = w_p & i = O(p), t = PT(p) \\ = -w_p & i = D(p), t = DT(p) \\ = 0 & o.w. \end{cases} \qquad \forall i \in \mathcal{V},\ t = 1,\dots,T,\ p = 1,\dots\mathcal{P} \tag{9}$$

$$U^{assign}_{pit} \begin{cases} \le S_i & S_i > 0, t = 0 \\ = -w_p & i = O(p), t = PT(p) \\ 0 & o.w. \end{cases} \qquad \forall i \in \mathcal{V},\ t = 1,\dots,T,\ p = 1,\dots\mathcal{P} \tag{10}$$

$$U^{repos}_{pit} \begin{cases} = w_p & i = D(p), t = DT(p) \\ \ge -S_i & S_i > 0, t = T \\ 0 & o.w. \end{cases} \qquad \forall i \in \mathcal{V},\ t = 1,\dots,T,\ p = 1,\dots\mathcal{P} \tag{11}$$

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 $S_i > 0$. Their destination locations are the locations that they are needed to be loaded and transport the products (O(p),PT(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)). Constraint (12) enforces the number of empty RTIs at a storage locations at the beginning of the planning horizon to be equal to the number at the end of the planning horizon.

$$\sum_{j\in\mathcal{V}-\{i\}} \sum_{m=1}^{\mathcal{M}+1} \sum_{p=1}^{\mathcal{P}} \check{x}^{assign}_{p,(i,j),0,(m_1,m_2)} = \sum_{j\in\mathcal{V}-\{i\}} \sum_{m=1}^{\mathcal{M}+1} \sum_{p=1}^{\mathcal{P}} \hat{x}^{repos}_{p,(j,i),T,(m_1,m_2)} \qquad \forall i \in \mathcal{V} : S_i > 0 \tag{12}$$

Constraints (13) and (14) 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 ($b = 0$).

$$\check{x}^{laden}_{p,(i,j),t,(m_1,m_2)} \ge \check{b}_{p,(i,j),t,(m_1,m_2)} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V}\times\mathcal{V}),\ p = 1,\dots\mathcal{P} \tag{13}$$

$$\check{x}^{laden}_{p,(i,j),t,(m_1,m_2)} \le M\check{b}_{p,(i,j),t,(m_1,m_2)} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V}\times\mathcal{V}),\ p = 1,\dots\mathcal{P} \tag{14}$$

Based on Constraints (13) and (14), Constraint (15) states that for each order, the total *time* × *temperature* of moving and handling an order must be less than or equal to the total required TTS of that order.

$$\sum_{a_{(i,j),t,(m_1,m_2)}\in A_1} l^{m}_{(i,j),t} \times r^{m}_{(i,j),t} \times \check{b}_{p,(i,j),t,(m,m)} + \sum_{a_{(i,j),t,(m_1,m_2)}\in A_2} l^{(m_1,m_2)}_{i,t} \times r^{(m_1,m_2)}_{i,t} \times \check{b}_{p,(i,i),t,(m_1,m_2)} \le \mathcal{L}_p \qquad \forall p = 1,\ldots \mathcal{P} \tag{15}$$

Constraint (16) is the capacity constraint. The total number of RTIs (laden or empty) that is moved between i and j at time t should be less than or equal to the total capacity of $\check{y}$ vehicles of mode m transporting them, if that mode is chosen.

$$\sum_{p=1}^{\mathcal{P}} \check{x}^{\text{laden}}_{p,(i,j),t,(m_1,m_2)} + \sum_{p=1}^{\mathcal{P}} \check{x}^{\text{assign}}_{p,(i,j),t,(m_1,m_2)} + \sum_{p=1}^{\mathcal{P}} \check{x}^{\text{repos}}_{p,(i,j),t,(m_1,m_2)} \le cap^m \times \check{y}_{(i,j),t,(m_1,m_2)} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A_m \tag{16}$$

Let $A_{t,m} = \{a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V}\times\mathcal{V}) \mid m_1 = m, m_2 = m, \tilde{t} \le t, \hat{t} \ge t\}$ be the set of all arcs of mode $m \in \{1,\ldots,\mathcal{M}\}$ crossing time period t. Constraint (17) 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^m.

$$\sum_{a_{(i,j),t,(m_1,m_2)}\in A_{t,m}} \check{y}_{(i,j),t,(m_1,m_2)} \le |F^m| \qquad \forall t = 1,\ldots,T,\; m = 1,\ldots,\mathcal{M} \tag{17}$$

Finally, Constraints (18)–(20) define the nature of the variables in our formulation.

$$\check{x}^{\text{cat}}_{p,(i,j),t,(m_1,m_2)},\; \hat{x}^{\text{cat}}_{p,(i,j),t,(m_1,m_2)} \ge 0 \qquad \forall cat \in \{laden, empty, repos\},\; a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V}\times\mathcal{V}),\; p = 1,\ldots \mathcal{P} \tag{18}$$

$$\check{b}_{p,(i,j),t,(m_1,m_2)},\; \hat{b}_{p,(i,j),t,(m_1,m_2)} \ge 0 \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A_1,\; p = 1,\ldots \mathcal{P} \tag{19}$$

$$\check{y}_{(i,j),t,(m,m)},\; \hat{y}_{(i,j),t,(m,m)} \in \mathbb{N} \qquad \forall a_{(i,j),t,(m_1,m_2)} \in A(\mathcal{V}\times\mathcal{V}) \tag{20}$$

Constraints (10), and (11) and their related flow conservation constraints are new to the literature of FCMNFP problems. Constraints (13), (14), and (15) which represent product quality requirements are also new to the literature of FCMNFP problems. In the next section, we describe the proposed solution algorithm in detail.

4. Solution algorithm

The biggest challenge in solving our multimodal network flow problem is the size of the model. In other words, even for small instance sizes, the number of variables is huge. Any state-of-the-art MIP solver (e.g. CPLEX, Gurobi) needs huge processing memory to deal with these variables. As already mentioned in Section 1, this problem is NP-hard, therefore in this section, we present a solution algorithm that handles size-related issues while preserving the solution quality.

The proposed solution method is an ALNS algorithm which starts with constructing an initial solution and tries to improve this solution in an iterative improvement phase. Meantime, the feasibility of the generated and modified solutions regarding the capacity of the modes, the maximum vehicle constraint, and the order TTS are guaranteed.

To understand the performance of this algorithm, in the following sections, we explain the phases step-by-step, starting with some definitions.

4.1. A solution and the solution space

A solution s consists of a set of *routes* and a set of *fleet*. A route r_p represents the scheduled flow of (laden or empty) RTIs in an order p. In other words, an r_p is the set of all $\check{x}^{\text{laden}}$ (or $\check{x}^{\text{assign}}$, or $\check{x}^{\text{repos}}$) with positive values. It shows the sequence of the locations that the flow of an order passes through, as well as the departing schedules of the modes which transport the RTIs among these locations. The $f^m_{(i,j),t}$ in the set of fleet represents the mode m used for the arc (i, j) departing at time t. The set of fleet shows the number of vehicles used ($\check{y}$ with positive values) for each space-time-mode arc. As a result, the solution space is the set of all possible routes and their related fleet arrangements.

4.2. Construction of an initial solution

To build an initial solution s_0, not only the routes of laden flows should be planned, but also the empty RTIs should be assigned and repositioned. Therefore, the algorithm first generates *empty orders*, which define how many empty RTIs and between which locations and at what time should be transported. After adding them to the set of all orders, the algorithm builds the routes. Fig. A.7 in the Appendix shows the initial solution generation process.

Exact and heuristic assignment algorithms

In order to assign empty RTIs to the orders, and later, reposition and return them back to their initial storage, we solve an assignment problem modeled in (21)–(24) twice. In the first, we have a number of supply locations $\bar{S}$ and a number of demand locations $\bar{D}$. Supply locations are the RTI storage locations with available number of RTIs $\bar{s}_i \geq 0$, and the demand locations are the origin points $O(p)$ with demand $\bar{d}_j = w_p$. In the second though, supply locations are the destination points $D(p)$ with supply $\bar{s}_i = w_p$ and the demand points are the storage locations with RTI deficit $\bar{d}_j \geq 0$. $\bar{x}^m_{(i,j),(t_1,t_2)}$ is the variable for the number of RTIs assigned from location i to location j with the pickup time t_1 and delivery deadline of t_2, transported by mode m, $\bar{r}^m_{(i,j),t_1}$ is the travel time by mode m between location pairs i and j starting at time t_1, and the objective function is to minimize total travel times.

$$\min \sum_{i=0}^{\bar{S}} \sum_{j=0}^{\bar{D}} \sum_{m \in \mathcal{M}_m} \bar{r}^m_{(i,j),t_1} \bar{x}^m_{(i,j),(t_1,t_2)} \tag{21}$$

S.T.

$$\sum_{j=0}^{\bar{D}} \sum_{m \in \mathcal{M}_m} \bar{x}^m_{(i,j),(t_1,t_2)} = \bar{s}_i \qquad \forall i = 0, \dots, \bar{S} \tag{22}$$

$$\sum_{i=0}^{\bar{S}} \sum_{m \in \mathcal{M}_m} \bar{x}^m_{(i,j),(t_1,t_2)} = \bar{d}_j \qquad \forall j = 0, \dots, \bar{D} \tag{23}$$

$$\bar{x}^m_{(i,j),(t_1,t_2)} \geq 0 \qquad \forall i = 0, \dots, \bar{S},\; j = 0, \dots, \bar{D},\; m \in \mathcal{M}_m \tag{24}$$

The abovementioned assignment problem is an extension to the classic transportation problem which can be solved via a state-of-the-art MIP solver in a polynomial time. However, for instances with large number of orders, a *bi-directional fastest fit* heuristic is used. Fig. A.6 in the Appendix shows the bi-directional fastest fit algorithm. This heuristic is a greedy algorithm where for each order, it tries to find the fastest possible assignment of empty RTIs. Note that the fastest possible assignment is used to ensure a feasible initial solution. We sort the order list based on volume, but to avoid a low utilized fleet, instead of choosing orders in a descending or ascending fashion, at the end of each iteration, we reverse the order list and assign them in a bi-directional fashion. Later in Section 5, we compare the performance of the bi-directional heuristic with the exact assignment.

Earliest-cheapest routing algorithm

After solving the assignment problem twice, we add the new empty orders to the set of all orders. Then, we run a fast *earliest-cheapest routing* algorithm to construct an initial route for each order. This algorithm is a *breadth-first search tree* based on the available modes, and its objective is to find the cheapest path with the earliest possible schedule from each origin to each destination at the defined pickup time. In order to pick an order, there are two options: either to choose an order randomly, or to give more priority to the orders with tighter schedules. Later in Section 5, we show the influence of this randomness on the quality of the final solution.

4.3. Adaptive Large Neighborhood Search

After constructing an initial solution, we use an ALNS search engine to improve the solution iteratively. ALNS was first introduced by (Ropke and Pisinger, 2006) and used in solving various problems (Section 2). We use a similar framework here. We start with describing our operators used to search neighborhoods, then we explain the weighting mechanism of our ALNS. Afterwards, we introduce the extra strategies that we used to improve our ALNS.

Operators and the solution evaluation

In our ALNS, there are two sets of *destroy* and *repair* operators. Each set represents a specific neighborhood, and at each iteration, the ALNS chooses one destroy and one repair operator to search these neighborhoods and to find improvements. A neighborhood of the current solution s is the set of all points in the solution space which can be reached by modifying the route of one or several orders and the resulting change in the fleet arrangement.

We initially designed exact operators which would remove particular parts of the solution and solve a standard NFP with modified right-hand side values. However, the number of variables, constraints, and the computational burden still remained high, even for the smallest neighborhoods. Later, we tried operators with classic *node-based* and *arc-based* neighborhoods of Ropke and Pisinger (2006). Despite their low computational effort, the improvement was not satisfactory at all. There

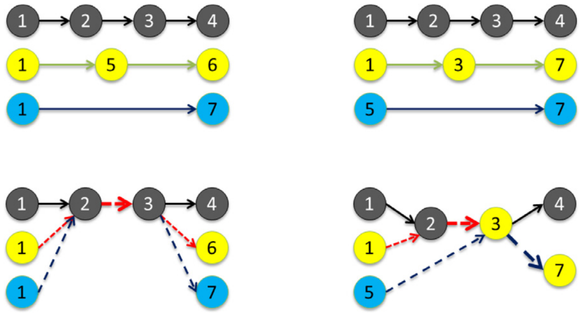


Fig. 4. An example of a consolidation option in the *consolidate three order* neighborhood; left: consolidate three orders (single), right: consolidate three orders (paired).

are also *cycle-based* neighborhoods in the literature (Ghamlouche et al., 2003) which have shown great performance, but they work only where assets have cyclic routes. Based on the description of our problem in Section 3, we chose path-based neighborhoods for our operators. The designed operators are new to the ALNS literature and we explain them in the following section.

We choose the following destroy operators which remove the routes of orders: *one random order* (R1) operator, *two random orders* (R2r) operator, *three random orders* (R3r) operator, *two similar orders* (R2s), and *three similar orders* (R3s) operator. The chosen orders can be laden orders, empty orders, or the combination of both. The number of possibilities in a random category is equal to the size of order set, while the number of combinations in R2s and R3s categories depends on the composition of the order set P. In the random category, the orders are chosen randomly. However, in the operators with similar orders, the orders are chosen based on two main similarities: their origin locations $O(p)$ (or their destination locations $D(p)$) should be equal, and the difference in their pickup times $PT(p)$ (or their delivery time $DT(p)$) should be less than β time periods. The reason for β is to avoid investigating orders that have completely different time and TTS requirements, as their chance to be consolidated together is low.

We have the following repair operators: *cheapest path* (Ip) operator, *scheduling* (Is) operator, *consolidate two orders* (Ic2), *consolidate three orders (single)* (Ic3s), and *consolidate three orders (paired)* (Ic3p) operator. The cheapest path operator has a similar structure to the Dijkstra' classic shortest path algorithm, but instead of time, the cost of the path is evaluated.

The scheduling operator tries to find feasible consolidation options by putting RTIs on later schedules of the used modes. For this purpose, two extra measure were created: the *earliest depart time* of the mode (ED^m_{ij}) and the *latest depart time* of the modes (LD^m_{ij}). These two values represent the feasible slack time of the departing time of each mode. These slack times are first collected for all modes in the fleet at the beginning of the improvement phase, and each time that a new mode is used in any iteration, its slack times are recomputed.

Finally, the consolidation operators aim to find the cheapest possible consolidation of the destroyed routes. In this regard, first a subnetwork is generated based on all the locations on the routes of these orders. Then, for each order, a set of all possible paths on the subnetwork with their departing slack times are generated. Later, all the combinations of paths of orders are compared in order to find any possibility where the mode on the paths are similar and their departing slack times overlap. Among these possibilities, the cheapest one is chosen. There is a fundamental difference between the consolidation of three order neighborhoods Ic3s and Ic3p. Fig. 4 gives an illustration of this difference. 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 other neighborhoods are capable of finding.

At the end of each iteration, the repaired routes are inserted into the current solution. Note that to evaluate the objective at each iteration, only cost changes are computed. In fact, at each iteration i we look at $\Delta(f_i)$ defined as $f_{Rep}(r_q) - f_{Des}(r_q)$ where $f_{Des}(r_q)$ is the total cost of the $|q|$ destroyed routes and the related removed items in the fleet set, and $f_{Rep}(r_q)$ is the total cost of the $|q|$ repaired routes and the added items in the fleet set.

Weighting mechanism

Similar to the classic ALNS, at the beginning of the improvement phase, all the neighborhood operators have weights $w_{i,0}$ equal to one. At each iteration and by looking at the weights, one destroy and one repair operator are chosen based on a roulette wheel (Ropke and Pisinger, 2006). At the end of the iteration e, depending on the performance of the chosen operator and whether it is able to improve the current and the best found solution, its weight is updated:

$$w_{i,e+1} = \lambda w_{i,e} + (1-\lambda)\theta \times \Delta(f_i) \tag{25}$$

where θ is a score θ_1 if the operator i results in improving the global best solution, a score θ_2 if it improves the current solution, a score θ_3 if the operator returns a feasible solution and accepts it, and a score θ_4 if the operator reject the

Table 1
Transportation mode inputs.

m	F^m	cap^m	l^m	$freq^m$	$speed^m$	C^m_{fix}	C^m_{full}	C^m_{empty}	
Truck services	200	22	5	1	65.00	136.34	6.412	2.849	2
Train services	100	1,760	5	6	32.50	179.37	0.025	0.011	2
Barge services	100	704	5	2	18.52	118.04	0.008	0.004	2

solution. The coefficient λ is a reaction factor which controls the reaction to the changes in the performance of the algorithm (Ropke and Pisinger, 2006). If it is equal to zero, the scores are not used, and if it is equal to one, the scores decide upon the weights. This weighting system gives the ANLS the opportunity to intensify the search by giving the operators with higher weights more chance to be chosen, while it tries to diversify the search by the roulette wheel mechanism.

The ALNS engine of our algorithm runs until a stopping criterion is met. Fig. A.8 in the Appendix shows these steps. We choose a stopping criterion based on the ρ number of iterations that the best found solution is not improved. However, in order to diversify our search and skip repetitions, we add some memory and diversification strategies.

4.4. Tabu memory lists

In the destroy operators that look for similar orders, we use a random seed to pick the first order. Therefore, randomness plays a big role in the destroy operators. In order to decrease the influence of this randomness on computational time, for each repair operator, we keep a short-term memory (a tabu list) with the size related to $min(\alpha, \binom{|O_s|}{|q|})$. The value $|O_s|$ is the size of the order list in a solution s, and $|q|$ is the number of removed orders. At each iteration, if the combination of these orders does not exist in the memory list, we add it to the list. Otherwise, we remove another set of orders. This strategy is particularly useful when the size of the problem increases and we like to search more parts of the solution space.

4.5. Diversification strategies

We want to make sure that we search the solution space as good as possible. Therefore, in addition to keeping short-term memory, we also keep a long-term memory, tracking the number of times each order has been chosen in all iterations. Later, we sort our order set based on these statistics in an ascending order and replace the standard destroy operators of the ALNS with the followings: *one last order* (D1) operator, *two last orders* (D2l), *three last orders* (D3l), *two last similar orders* (D2s), and *three last similar orders* (D3s) operator. Clearly, the first three operators pick the orders with the least number of times that they have been checked during all iterations. The later three operators on the other hand, try to choose the least checked similar orders. We then run the ALNS with the same repair operators again to find further improvements. This diversification step also has a stopping criterion of $\gamma = 0.5\rho$ number of iterations without any improvements.

In the next section, we present the computational results and compare the performance of our ALNS metaheuristic with a state-of-the-art MIP solver.

5. Computational results

In this section, we test the algorithm presented in Section 4 on different instances. This research is inspired by the transportation of horticultural products on the Trans-European Transport Network (TEN-T) (Tosi, 2014; Verhoeven, 2014; Vlassak, 2014; Rosenboom, 2014). Fig. 5 shows the 20 hubs where two of them are inbound hubs and the others are outbound hubs. Inbound hubs are locations where the products come from all around the world to get sold, sorted, and consolidated for the shipments. The outbound hubs on the other hand are locations where the shipments are divided and packaged for the last-mile distribution. We categorize our instances in three groups of 7, 11, and 20 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.

Regarding the transportation modes, we consider three different means of transport: truck, train, and barge. Tables 1 gives the parameter setting for the modes. These numbers have been extracted from company documents and annual EU reports on transportation (Tosi, 2014; Verhoeven, 2014; Vlassak, 2014; Rosenboom, 2014). In this table, F^m and cap^m show the maximum number of available vehicles and their capacity for each mode type. l^m is the given temperature for each mode type based on °C. $freq^m$ 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, C^m_{laden}, and C^m_{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. As we explained in Section 1, multimodal transportation is increasing, therefore, for the initial experiments, we assume to have a complete multimodal network in order to see what modes are utilized in a restriction-free system. Later, we compare them with the case of a real-world network.

For orders, we again used the actual demand and supply of each location based on the annual numbers (Rosenboom, 2014), and aggregated them to fit our setup. These orders have 12 hour, 24 hour, or 48 hour delivery service, and their TTS is 200 hour-degree.

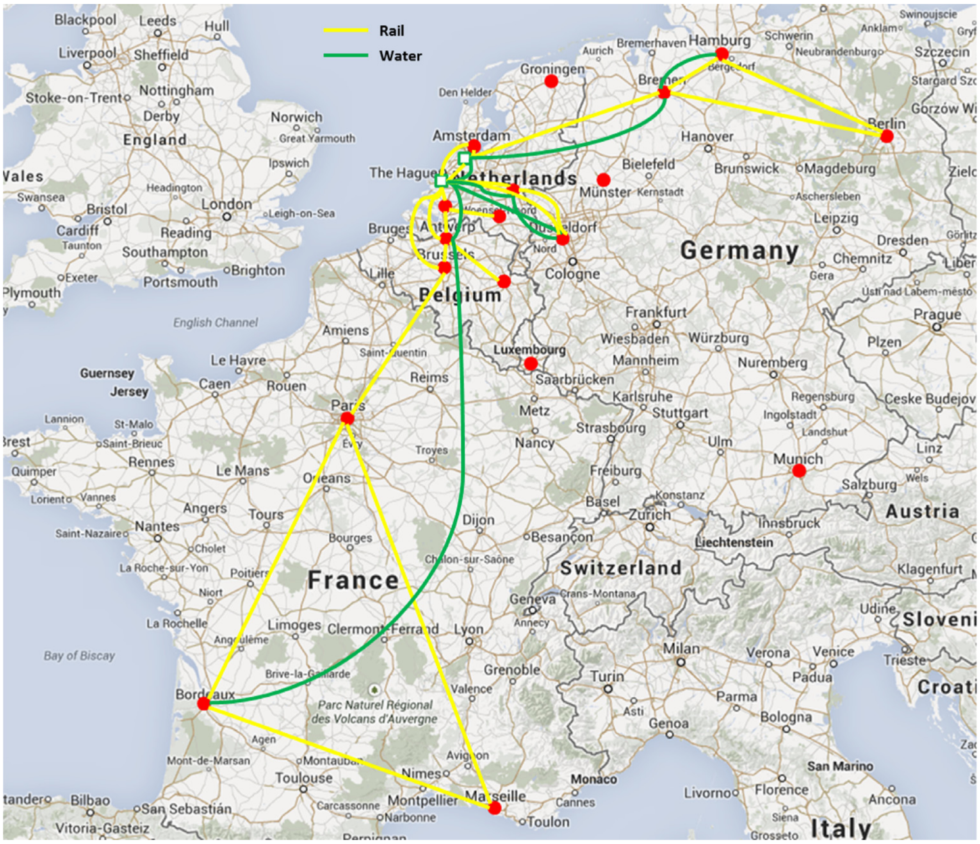


Fig. 5. The horticultural network of Europe with its real-world rail and water connections; the squares are the inbound hubs and the circles show the outbound hubs;.

The empty RTIs in this network are stored at the inbound hubs. Due to the fact that the repositioning of medium-sized RTIs (trolleys) is critical in the horticultural chain, we choose them as our RTI in this paper and assume to have a total of 30,000 at the inbound hubs.

The instances are named as "nAmBtCoDE" where the value of A shows the number of locations, value of B is the number of mode types, value of C is the planning horizon, and value of D stands for the number of orders. The instances with $m1$ only have truck options, the instances with $m2$ have truck and train options, and the instances with $m3$ have all modes available. If we have an order set with tight delivery due dates, E is "t", and if the delivery due dates are loose, E is "l". We solve all instances on a 2.4 GHz CPU with 8.00 GB RAM and we use Gurobi solver 6.5.0 as our MIP solver.

In the following sections, we describe the parameter setting, give the computational results, and provide some sensitivity analysis.

5.1. Parameter tuning

Prior to showing the final results and discussing the quality of our ALNS solutions, we discuss its tuning. For the basic parameters, after several initial tests, we chose the following values: ALNS scoring parameters of $\theta_1 = 30.0$, $\theta_2 = 20.0$, $\theta_3 = 10.0$, $\theta_4 = 1.0$, and $\lambda = 0.5$, the time interval of $\beta = 10$ and the stopping criteria of $\rho = 500$ and $\gamma = 125$. Higher stopping criteria might improve the solution even further, but the improvement compared to the time spent on the extra iterations does not seem useful. Higher values for θ_1 and θ_2 also might result in getting stuck in local optima and less diversified neighborhood search.

5.2. Overall results

In this section, we solve the model with the Gurobi solver and compare its results with the ones from our proposed ALNS. In the smallest generated instance n7m1o10, there are 3222 arcs, and 25,776 decision variables with positive cost coefficients in the objective function. This is already larger than the majority of the existing instances in the literature. Keeping all the matrices can cost a huge amount of memory. The matrices of parameters and variables are extremely sparse, so in order to

Table 2
Overall results of ALNS.

				Exact (10 hours)				Exact (15 min)			ALNS metaheuristic (in 15 min)		
n	m	t	o	Lower bound	Upper bound	Gap (%)	Comp. time (sec.)	Lower bound	Upper bound	Gap (%)	Best found total cost	Diff. (%) exact 10 hr	Diff. (%) Exact 15 min
n7	m1	t48	o10	55,164	55,167	0.00	4,423	–	–	–	55,167	0.00	-
n7	m2	t48	o10	13,787	13,786	0.00	1,018	12,815	13,786	3.96	14,071	2.07	2.07
n7	m3	t48	o10	13,263	13,264	0.00	3,269	12,285	–	–	13,497	1.76	-
n7	m1	t48	o25	130,592	131,248	0.50	10H	129,732	–	–	131,510	0.20	-
n7	m2	t48	o25	31,109	31,112	0.00	4,293	29,721	–	–	32,195	3.48	3.45
n7	m3	t48	o25	30,499	30,501	0.00	29,219	29,653	330,878	4.00	31,031	1.74	−0.12
n7	m1	t48	o50	248,956	249,644	0.21	10H	247,605	–	–	250,672	0.41	-
n7	m2	t48	o50	60,989	60,992	0.00	23,090	59,233	61,235	2.05	62,246	2.06	1.65
n7	m3	t48	o50	60,159	60,185	0.04	10H	–	60,967	3.10	62,042	3.09	1.76
n7	m3	t48	o100	117,911	118,265	0.35	10H	–	121,818	4.41	121,320	2.58	−0.41
n7	m3	t48	o200	227,151	228,223	0.43	10H	–	–	–	236,215	3.50	-
n11	m1	t48	o10	14,492*	14,879*	2.66	10H	13,936	–	–	14,879	0.00	-
n11	m2	t48	o10	4,507	4,507	0.01	5,309	3,724	4,510	21.10	5,089	12.90	12.84
n11	m3	t48	o10	3,313	3,775	13.93	10H	2,962	3,931	32.71	4,231	12.08	7.62
n11	m1	t48	o25	40,526	40,827	0.74	10H	38,327	40,827	6.52	41,420	1.45	1.45
n11	m2	t48	o25	14,336	14,522	1.30	10H	12,654	14,865	17.47	15,807	8.85	6.33
n11	m3	t48	o25	13,708	13,708	0.00	19,912	12,064	14,023	16.24	14,386	4.94	2.59
n11	m1	t48	o50	80,814	81,463	0.80	10H	78,654	–	–	82,450	1.21	–
n11	m2	t48	o50	32,183	32,185	0.00	10,844	28,679	32,436	14.34	34,998	8.74	7.90
n11	m3	t48	o50	30,027	31,507	4.9	BB	24,131	–	–	32,253	2.42	–
n11	m3	t48	o100	60,985	62,862	3.07	10H	–	–	–	65,985	4.97	–
n11	m3	t48	o200	118,019*	122,292*	3.61	10H	–	–	–	135,430	10.74	–
n20	m1	t72	o10t	–	–	–	Rx	–	–	–	57,370	–	–
n20	m2	t72	o10t	19,942	–	–	BB	–	–	–	22,273	–	–
n20	m3	t72	o10t	19,688	–	–	BB	–	–	–	20,976	–	–
n20	m1	t72	o25t	143,194	–	–	BB	–	–	–	147,624	–	–
n20	m2	t72	o25t	51,428	–	–	BB	–	–	–	55,895	–	–
n20	m3	t72	o25t	–	–	–	Rx	–	–	–	56,274	–	–
n20	m1	t72	o50t	–	–	–	Md	–	–	–	291,816	–	–
n20	m2	t72	o50t	–	–	–	Rx	–	–	–	108,454	–	–
n20	m3	t72	o50t	–	–	–	Rx	–	–	–	103,338	–	–
n20	m3	t72	o100t	–	–	–	Rx	–	–	–	174,216	–	–
n20	m3	t72	o200t	–	–	–	Md	–	–	–	385,521	–	–
n20	m1	t72	o10l	54,778	–	–	BB	–	–	–	57,371	–	–
n20	m2	t72	o10l	–	–	–	Rx	–	–	–	12,514	–	–
n20	m3	t72	o10l	8,741	10,353	18.43	10H	–	–	–	11,724	13.24	–
n20	m1	t72	o25l	–	–	–	Rx	–	–	–	148,426	–	–
n20	m2	t72	o25l	14,920	–	–	BB	–	–	–	21,434	8.67	–
n20	m3	t72	o25l	11,306	–	–	BB	–	–	–	18,421	0.50	–
n20	m1	t72	o50l	–	–	–	Rx	–	–	–	289,758	–	–
n20	m2	t72	o50l	–	–	–	Md	–	–	–	44,215	–	–
n20	m3	t72	o50l	–	–	–	Md	–	–	–	40,158	–	–
n20	m3	t72	o100l	–	–	–	Md	–	–	–	73,893	–	–
n20	m3	t72	o200l	–	–	–	Md	–	–	–	178,640	–	–
n20	m2	t72	o1000t	–	–	–	Md	–	–	–	483,754	–	–
n20	m3	t72	o2000t	–	–	–	Md	–	–	–	502,649	–	–
n20	m3	t72	o1000l	–	–	–	Md	–	–	–	156,344	–	–
n20	m3	t72	o2000l	–	–	–	Md	–	–	–	401,739	–	–

Note: 10H is an abbreviation for 10 hour time.
Note: Md means that Gurobi ran out of memory in building the MIP model, Rx means that Gurobi ran out of memory while solving the relaxation of the problem, and BB means that Gurobi ran out of memory during the branch-and-bound search, within 10 hours (BB).
* These results have been obtained by setting a MIP initial solution equal to the initial solution of the ALNS.

decrease the memory consumption of the Gurobi solver, we use the so-called "colt"[1] library to replace the standard matrix format with a sparse one. We run the Gurobi solver for a maximum duration of 10 hours.

The first three columns of Table 2 give the obtained lower bounds, upper bounds, and the optimal gap within 10 hours. We are not able to find the optimal solution of the majority of the instances in 10 hours. In the group of $n20$, we are almost unable to find a lower bound. The computation time depends not only on the number of orders, but on the number of modes and the tightness of delivery times. It is harder to find a solution for $m1$ instances than for $m2$ and $m3$. For those

[1] http://acs.lbl.gov/software/colt/.

Table 3
Comparison of the performance of ALNS with different repair combinations.

n	m	t	o	S0 Average total cost	S0 A.D. (%)	S1 Cost difference	S1 A.D. (%)	S2 Cost difference	S2 A.D. (%)	S3 Cost difference	S3 A.D. (%)	S4 Cost difference	S4 A.D. (%)
n7	m3	t48	o100	125,511	0.17	−1,220	0.22	−2,278	0.35	−2,707	0.10	−**3,424**	0.50
n11	m3	t48	o10	4,643	1.07	−**444**	0.63	−372	1.78	−439	0.91	−423	4.23
n11	m2	t48	o25	17,244	0.70	−1,004	1.22	−1,455	1.63	**−1,965**	0	−1,466	1.83
n11	m3	t48	o200	145,096	0.01	−3,297	0.38	−4,960	1.14	−4,547	0.67	−**7,667**	0.25
n20	m3	t72	o50t	108,900	0.46	−5,339	0.78	−4,260	0.10	−4,128	1.29	−**3,242**	0.92
n20	m3	t72	o100t	180,420	0.19	−5,631	0.33	−**7,633**	0.53	−6,222	0.54	−7,576	0.35
n20	m3	t72	o10l	13,889	0.80	−2,117	0.61	−2,058	0.19	−2,071	0	−**2,125**	0.00
n20	m3	t72	o25l	21,571	0	−2,517	0	−2,131	3.08	−**3,111**	0.44	−2,738	3.92
n20	m3	t72	o200l	203,215	2.27	−20,080	0.06	−23,839	1.35	−25,510	0.90	−**29,209**	2.13
		Average		–	0.63	–	0.47	–	1.13	–	0.54	–	1.57

S0: including repair operators Ip, Is.
S1: including repair operators Ip, Is, Ic2.
S2: including repair operators Ip, Is, Ic2, Ic3p.
S3: including repair operators Ip, Is, Ic2, Ic3s.
S4: including repair operators Ip, Is, Ic2, Ic3p, Ic3s.

instances with no bounds, despite the computational trick we use, Gurobi solver runs out of memory during the branch-and-bound, during solving the relaxation of the problem, or during constructing the model itself. In an extra attempt, for those instances that Gurobi could construct the model, we gave the initial solution of ALNS as the MIP start solution, which only had a limited effect as we were able to find near-optimal solutions for only two more instances n11m1o10 and n11m3o200.

In order to compare the performance of our ALNS with the Gurobi solver, we run both of them for 15 minutes (Table 2). The last two columns of the table show the difference between the obtained upper bounds for both Gurobi runs and our ALNS. As shown in the table, Gurobi solver is not able to find a lower bound for about two third of the instances. Moreover, among the other instances, the ALNS solutions have an average optimality gap of 3.9% which is acceptable considering the size of our problem.

Looking at the total costs in Table 2, an important trend seen is that by involving trains, and further barges, the total cost decreases by more than 50%. These results encourage the usage of multimodal transportation.

5.3. Algorithm performance

In this section, we discuss the efficiency of the destroy and repair operators based on both the time spent and their share in improving the solution. We furthermore analyze strategies used to improve the results and reduce the computation time. We also check the impact of the quality of the initial solutions on the performance of the algorithm. To keep the length of this research reasonable, for our performance analysis, we chose nine out of all 48 instances. These sample instances have adequate variety in number of locations and orders.

Operator combinations

In order to analyze the behavior of the repair operators, we defined a few scenarios, each of which implements a particular combination of the repair operators. The number of possible combinations is high, but the scenarios we present in this section give a nice overview on the behavior of the operators. We chose to keep operators Ip and Is in all scenarios as they are fast and effective. Table 3 shows a comparison of the average total cost obtained and the average deviations (A.D.) over 10 repetitions of ALNS. As it is shown in Table 3, scenario *S*4 which includes all operators, has the best performance. Moreover, the performance of scenarios *S*2 and *S*3, especially in larger instances, show that the repair operators which consolidate three order routes are important contributors to the cost improvement. However, based on Table 4, these operators are time-consuming. This is due to the fact that the consolidation operators need to analyze large sets of consolidation options.

In order to compare the behavior of the repair operators, we included all our destroy operators in each scenario (in Tables 3 and 4). Now, we study the role of each destroy operator. For this purpose, we used scenario S4 (with all the repair operators). Since the destroy operators have a similar structure and mostly the number of orders they remove is different, the scenarios with different combinations of these operators are not that distinct. Therefore, we ran scenario S4 once including each of the operators alone, and once excluding them, and we checked how much on average the cost of the solutions are different from the cost of the solution of scenario S4 including all of the destroy operators.

Table 5 shows the role of each of the destroy operators over 10 repetitions of our ALNS. First of all, as shown in Table 5, the time these operators spend to remove a set of orders is a matter of milliseconds. Next, by looking at the table, we can see that excluding each of them results in more expensive solutions by around 1%. It is even more expensive to use them alone. In conclusion, each destroy operator has a significant contribution to the cost improvement.

Table 4
Comparison of the computation time (Seconds) of ALNS with different repair combinations.

n	m	t	o	S0	S1	S2	S3	S4
n7	m3	t48	o100	2.00	4.87	8.19	104.83	100.24
n11	m3	t48	o10	0.78	1.60	2.85	21.26	59.98
n11	m2	t48	o25	0.84	1.23	2.72	13.89	5.60
n11	m3	t48	o200	9.24	12.19	30.55	83.19	387.31
n20	m3	t72	o50t	3.63	12.25	84.41	1,775.01	2,723.86
n20	m3	t72	o100t	8.49	33.92	202.51	1,226.20	5,798.21
n20	m3	t72	o10l	0.85	1.67	6.39	19.36	35.86
n20	m3	t72	o25l	1.62	3.66	16.39	352.43	410.20
n20	m3	t72	o200l	25.91	124.11	1,196.12	2,570.61	7,968.14

S0: including repair operators Ip, Is.
S1: including repair operators Ip, Is, Ic2.
S2: including repair operators Ip, Is, Ic2, Ic3p.
S3: including repair operators Ip, Is, Ic2, Ic3s.
S4: including repair operators Ip, Is, Ic2, Ic3p, Ic3s.

Table 5
The role of destroy operator.

		D1	D2r	D2s	D3r	D3s
Extra cost (%)	including the operator only	9.92	1.83	7.93	4.81	4.10
Extra cost (%)	excluding the operator	0.31	0.13	1.49	0.69	1.42
computation time (Seconds)		0.001	0.035	0.128	0.145	0.700

Scoring mechanism

In order to have an algorithm which includes the time-consuming consolidation operators while keeping the computation time low, we included the computation time of each iteration into the ALNS scoring system. Initially in all scenarios, the weights of the operators were updated based on $\Delta(f_i)$. In a new test (scenario $S5$), we changed the scoring formula to:

$$w_{i,e+1} = \lambda w_{i,e} + (1-\lambda)\theta \times \Delta(f_i)/\Delta(time)_e \tag{26}$$

The reason is to give less weight to the more time consuming operators by penalizing them by $\Delta(time)_e$. $\Delta(time)_e$ is the time it takes for the repair process. The results of this test are given in Table 6. By penalizing the time consuming operators, we not only could save on average 37.22% of the computation time, but also improve the average total costs over 10 repetitions.

Table 7 gives an overview of the average share of each operator in the cost improvements and the average time (in seconds) they spend in both scenarios S4 and S5. There is no particular difference seen in their share of improvement. All operators play significant roles in improving the solution, particularly operator Ip which ranks first. Ic3p and Ic3s are the heaviest pieces in computation, but if we ignore them completely (like in $S1$ in Table 3), we would lose the quality of the solution by 1.81, −0.51, 2.93, 3.18, −1.98, 1.13, 0.07, 1.17, and 5.52 percent over the nine sample instances.

Table 6
Comparison of the performance of ALNS with different scoring variables.

				S4			S5		
n	m	t	o	Best found cost	Average total cost	Comp. time (s)	Best found cost	Average total cost	Comp. time (s)
n7	m3	t48	o100	**121,161**	122,086	100.24	121,649	**121,951**	16.75
n11	m3	t48	o10	**4,135**	**4,220**	59.98	4,250	4,439	75.29
n11	m2	t48	o25	**15,553**	15,779	5.60	15,597	**15,741**	9.64
n11	m3	t48	o200	135,405	137,429	387.31	**134,443**	**134,704**	116.04
n20	m3	t72	o50t	105,265	105,658	2,723.86	**103,430**	**104,332**	524.80
n20	m3	t72	o100t	171,940	172,845	5,798.21	**171,739**	**172,044**	950.99
n20	m3	t72	o10l	11,764	11,764	35.86	**11,637**	**11,637**	45.67
n20	m3	t72	o25l	18,832	**18,832**	410.20	**18,625**	19,267	186.13
n20	m3	t72	o200l	173,992	**174,006**	7,968.14	**173,805**	178,862	980.65

S4: including all repair operators with $\Delta(f_i)$.
S5: including all repair operators with $\Delta(f_i)/\Delta(time)_e$.

Table 7
Comparison of the performance of the operators.

n	m	t	o	S4 Share of improvement in total cost (%) Ip	Is	Ic2	Ic3p	Ic3s	Computation time (seconds) Ip	Is	Ic2	Ic3p	Ic3s
n7	m3	t48	o100	49.65	0.89	6.13	38.84	4.48	2	0	1	14	185
n11	m3	t48	o10	69.90	2.79	18.22	8.47	0.62	1	1	2	54	773
n11	m2	t48	o25	49.69	1.76	4.55	22.19	21.82	1	1	2	5	12
n11	m3	t48	o200	58.37	8.48	18.67	11.45	3.03	6	1	3	12	195
n20	m3	t72	o50t	47.88	1.29	11.75	27.76	11.32	8	1	3	498	7,106
n20	m3	t72	o100t	69.25	2.32	6.69	8.57	13.17	36	1	7	492	9,649
n20	m3	t72	o10l	90.34	0.78	3.53	4.68	0.66	4	1	1	82	139
n20	m3	t72	o25l	84.23	0.91	12.00	2.61	0.26	7	0	7	185	1,061
n20	m3	t72	o200l	71.89	1.81	14.05	2.06	10.19	7	1	7	495	7,532
				S5 Share of improvement in total cost (%)					Computation time (seconds)				
n	m	t	o	Ip	Is	Ic2	Ic3p	Ic3s	Ip	Is	Ic2	Ic3p	Ic3s
n7	m3	t48	o100	59.51	1.18	13.68	24.54	1.08	2	1	0	8	10
n11	m3	t48	o10	37.54	2.81	14.30	45.23	0.11	1	0	3	38	853
n11	m2	t48	o25	57.50	2.92	3.88	35.62	0.08	1	1	0	6	32
n11	m3	t48	o200	52.27	7.57	5.72	29.04	5.40	5	0	2	7	31
n20	m3	t72	o50t	55.91	1.76	21.71	7.98	12.64	6	0	8	89	3,682
n20	m3	t72	o100t	58.70	2.05	7.83	19.38	12.05	12	0	4	87	1,674
n20	m3	t72	o10l	82.15	0.55	2.28	14.39	0.64	4	0	2	49	125
n20	m3	t72	o25l	91.64	0.45	0.43	5.75	1.73	5	0	6	137	11
n20	m3	t72	o200l	34.29	1.21	3.93	40.55	20.03	21	0	4	79	278

S4: including all repair operators with $\Delta(f_i)$.
S5: including all repair operators with $\Delta(f_i)/\Delta(time)_e$.

Table 8
Comparison of the performance of ALNS with and without diversification.

n	m	t	o	S5 Best total cost	Average total cost	A.D. (%)	Comp. time (sec.)	S6 Best total cost	Average total cost	A.D. (%)	Comp. time (s)
n7	m3	t48	o100	121,648	121,951	0.50	16.75	**121,212**	**121,489**	0.20	30.53
n11	m3	t48	o10	4,250	4,439	4.23	75.29	**4,137**	**4,255**	2.27	26.50
n11	m2	t48	o25	15,597	15,741	1.83	9.64	**15,581**	**15,582**	0.01	8.66
n11	m3	t48	o200	134,442	134,704	0.25	116.04	**133,115**	**134,236**	0.79	80.49
n20	m3	t72	o50t	**103,430**	**104,332**	0.92	524.80	104,238	104,681	0.36	719.96
n20	m3	t72	o100t	**171,739**	**172,044**	0.35	950.99	172,353	173,136	0.41	0
n20	m3	t72	o10l	**11,637**	**11,637**	0	45.67	12,041	12,066	0.41	56.18
n20	m3	t72	o25l	18,625	19,267	3.92	186.13	**18,124**	**18,779**	4.95	782.19
n20	m3	t72	o200l	173,804	178,862	2.31	980.65	**171,704**	**172,428**	0.46	3,243.35
		Average		–	–	1.59	–	–	–	1.09	–

S6: S5 plus the diversification strategy.

Diversification

In order to push the algorithm further, we added a diversification strategy which replaces the original destroy operators (D1, D2r, D2s, D3r, and D3s) with new D1, D2l, D3l, D2s, and D3s operators which try to look into the routes with the least number of times they have been checked in the earlier iterations. We compare the results of our ALNS without diversification and with diversification in Table 8. Diversification increases the computation time by an average of 54%, even though in some instances this time decreases. For instances with normal or loose delivery due dates, ALNS with diversification improves both best found costs and the average total costs over 10 repetitions. In contrast, for instances with tight time windows, diversification does not improve the costs. This is probably due to the fact that the orders with tight due dates have less chance to be consolidated into slower and cheaper transportation modes. In overall, the comparison of the average deviations shows that ALNS with diversification has a more robust outcome.

Quality of initial solutions

To test the robustness of the ALNS, the quality of the initial solution itself is investigated. In Section 4.2, we introduced two exact and heuristic assignment algorithms for the assignment of empty RTIs in the initial solutions. We also explained

Table 9
Comparison of exact assignment and heuristic assignment.

n	m	t	o	S6 Average initial cost	S6 A.D. (%)	S6 Average total cost	S6 A.D. (%)	S7 Average initial cost	S7 A.D. (%)	S7 Average total cost	S7 A.D. (%)
n7	m3	t48	o100	291,079	5.94	121,489	0.20	240,753	5.44	**100,266**	0.86
n11	m3	t48	o10	9,867	21.40	**4,255**	2.27	8,357	10.60	5,152	0
n11	m2	t48	o25	33,657	9.87	**15,582**	0.01	38,064	8.51	19,503	0.27
n11	m3	t48	o200	254,989	2.07	134,236	0.79	232,369	2.83	**113,134**	1.52
n20	m3	t72	o50t	195,979	8.74	104,681	0.36	186,076	4.59	**92,386**	0.45
n20	m3	t72	o100t	337,400	1.65	**173,136**	0.41	380,021	4.10	191,214	3.66
n20	m3	t72	o200t	731,008	2.08	**385,521**	0.73	739,022	2.50	407,219	4.05
n20	m3	t72	o10l	34,476	17.80	12,066	0.41	24,423	18.76	**10,316**	0.75
n20	m3	t72	o25l	74,611	6.10	**18,779**	4.95	64,018	10.67	22,647	0.44
n20	m3	t72	o100l	246,670	6.87	**73,893**	4.52	1,279,967	4.32	77,906	5.11
n20	m3	t72	o200l	653,806	3.59	**172,428**	0.46	510,641	5.77	189,498	0.00
n20	m3	t72	o1000t	N.I.	–	–	–	766,434	0.50	**483,754**	6.10
n20	m3	t72	o2000t	N.I.	–	–	–	1,437,041	1.01	**502,649**	6.48
n20	m3	t72	o1000l	N.I.	–	–	–	649,983	1.88	**156,344**	0.00
n20	m3	t72	o2000l	N.I.	–	–	–	1,216,513	0.96	**401,739**	7.09
			Average	–	7.82	–	1.37	–	5.50	–	2.45

N.I. stands for No obtained Initial solution in 15 min.
S7: S6 with heuristic assignment of empty RTIs.

Table 10
An overview of the mode usage in the solutions.

	Average number			Average utilization by this sector (%)			Average usage (%)		
	truck	train	barge	truck	train	barge	truck	train	barge
o10	8	6	10	69.2	2.6	11.7	11.2	25.7	63.0
o25	20	11	18	77.1	5.1	14.9	11.8	29.4	58.8
o50	41	18	26	80.5	4.8	20.4	13.9	24.5	61.6
o100	72	31	38	84.8	6.4	24.9	14.2	29.9	55.9
o200	160	44	62	86.7	12.1	29.5	14.5	31.5	54.0
o1000	174	134	133	75.4	4.9	11.0	12.7	45.2	42.0
o2000	370	245	255	74.2	5.4	9.1	13.6	52.0	34.3

that in the heuristic assignment procedure, we either choose orders randomly or prioritize the ones with tighter delivery due date. Table 9 shows that for very large instances (with more than 1000 orders), solving the initial assignment of empty RTIs via a state-of-the-art solver is not able to obtain any initial solutions even in 15 minutes time limit. In contrast, the ALNS with the heuristic assignment result in worse solutions in smaller instances compared to the exact algorithm.

Summarizing remarks

We showed that each operator plays an important role in the overall improvement and the algorithm shows a robust behavior. However, we need to control the time the operators spend processing. This is done by a scoring system depending on both cost change and time consumption. In addition, a diversification strategy is added in order to explore more areas of the solution space.

In the following section, we take a further look at the structure of the solutions.

5.4. Some practical insights

Since the main share of the costs belongs to the fixed costs of using the modes, we are interested in the combination of fleet and the number of vehicles for each transportation mode. Table 10 presents the number of vehicles for each mode type and the average percentage that each vehicle is utilized by this sector. It shows that on average 52.8% of the time barges are used and on average 17.4% of each barge is utilized by this sector. In contrast, trucks and trains are used 13.1% and 34.0% of the time, while on average 78.2% and 5.9% of their vehicles are utilized by this sector respectively. This is due to cheap costs of barge compared to the other two modes. Overall, barge seems to be the most favorable means of transportation. However, in very large instances, trains have a bigger share of usage. They are expensive even though faster than barges, and only with large number of orders they are cost-efficient.

One important issue is that the percentage of utilization of barges and trains by this sector in all instances is low. This is due to the fact that the products have strict delivery requirements and these modes are mostly used for repositioning RTIs.

Table 11
Comparison of the results between complete and incomplete multimodal network.

				Complete network				Incomplete network			
n	m	t	o	Average initial cost	No. trucks	No. trains	No. barges	Average initial cost	No. trucks	No. trains	No. barges
n20	m3	t72	o100t	174,216	108	34	47	336,113	306	40	26
n20	m3	t72	o200t	385,521	238	55	68	808,214	701	54	38
n20	m3	t72	o100l	73,893	51	49	56	226,586	232	44	26
n20	m3	t72	o200l	178,640	142	65	84	545,628	504	60	30
n20	m3	t72	o1000t	483,754	266	132	125	772,169	693	74	79
n20	m3	t72	o2000t	502,649	519	216	215	1,518,313	1,307	84	96
n20	m3	t72	o1000l	156,344	81	137	141	613,086	616	75	57
n20	m3	t72	o2000l	401,739	220	277	295	1,327,066	1,245	95	82

Table 12
Comparison of the repositioning costs of the ALNS with the simple rule-based heuristic.

n	o	Rule-based heuristic	ALNS m1 cost diff. (%)	ALNS m3 cost diff. (%)
n7	o10	35,549	−22	−81
n7	o25	86,861	−24	−82
n7	o50	165,244	−24	−81
n7	o100	328,325	−25	−82
n7	o200	719,163	−25	−84
n11	o10	9,736	−24	−78
n11	o25	27,965	−26	−74
n11	o50	57,283	−28	−72
n11	o100	114,985	−24	−71
n11	o200	243,434	−25	−72
n20	o10	37,057	−23	−72
n20	o25	95,897	−23	−71
n20	o50	190,858	−24	−73
n20	o100	353,298	−24	−75
n20	o200	754,340	−26	−74
n20	o1000	1,051,639	−27	−77
n20	o2000	966,633	−26	−74
Average		-	−25	−77

5.4.1. Incomplete transportation network

So far, our results were based on the assumptions that we have a complete multimodal network, while in reality, not all modes are feasible for each pair of locations. The reason to work under such an assumption was to allow each mode of transportation an equal chance, and to see their involvement without considering network boundaries.

In this section, we compare the results with a realistic TEN-T network. Fig. 5 gives an illustration of the actual rail and water connections in Europe with much fewer multimodal options for instances of n11 and n20. Still, we assume that all locations are still connected by road which is reasonable (not shown them in figure).

Table 11 shows the huge difference between the total cost of the results between the complete and incomplete network. In the incomplete network, the costs and the number of trucks used increase by on average 175% and 303% respectively. The numbers of trains and barges on the other hand decrease by on average 27% and 53%.

Looking at the structure of the routes in both complete and incomplete network results, we see that direct road connections are mostly used. However, in the incomplete network results, there is more variety of multimodal routes with two and three transshipments, while in the complete network results, the number of transshipments is at most one and the varieties with more transshipments are rare.

5.4.2. Repositioning rule-based heuristic

In the horticultural industry, in order to ensure that enough RTIs are available at the two inbound locations, a rule is enforced which requires the used RTIs to be repositioned back to their original inbound hubs as soon as possible.

In this section, we replace our repositioning heuristic with a simple rule-based heuristic based on the current practice. Table 12 compares the obtained repositioning costs of this simple heuristic with the repositioning costs of the ALNS.

The second and the third columns of the table show the cost differences for a truck-only network ($m1$) and a multimodal network ($m3$). Looking at the third column, there is a significant repositioning cost reduction by switching to multimodal transportation. The simple rule-based heuristic causes more usage of direct truck-only transportation with less consolidation. However, considering a truck-only network, our ALNS is still able to provide on average 25% cheaper repositioning solutions than the rule-based heuristic. These results encourage the application of our ALNS in practice.

6. Conclusions

In this paper, inspired by the horticultural supply chain of the Netherlands, we developed a MIP model for the multi-modal network flow problem for transporting perishable products, taking product quality preservation, and asset management into account. As such, we include the role of RTI allocation and repositioning in the optimal flow of products. This problem is NP-hard. We then proposed an ALNS algorithm to solve it. We introduced new path-based neighborhoods which at each iteration work on various combination of orders. We furthermore used specific scoring to manage the computation time of the algorithm. The ALNS uses a short-term Tabu list to add more diversification and prevent redundancies, but we also added an extra long-term memory to exploit unvisited parts of the solution space by looking at the neighborhoods of order routes that have not been investigated often.

We tested our model on 48 different instances with different number of locations, on a real network with up to 3 transportation modes, and sets of up to 2000 orders. The proposed algorithm is fast, does not take up much memory, and provides good solutions. It is even able to find solutions for instances that Gurobi solver cannot find any lower bounds for.

Possible future work can be to improve the assignment heuristic, for instance by adding an iterative local search which can improve the obtained assignment. The proposed ALNS in this paper was designed to solve a tactical planning level problem. Another future extension is to study the operational planning level problem, where this assumption is relaxed that the information on all coming orders is available, and via a rolling horizon planning framework, the solution algorithm responds to new information. This research is forthcoming.

Acknowledgement

The study in this article has been supported by the DaVinc3i project with the reference number 2010.2.034R. DaVinc3i is co-financed by Dinalog and the Horticultural Commodities Board. Moreover, we would like to thank the following partners for their support and contribution: FloraHolland, VGB, Kwekerij A. Baas, Bunnik Plants, DGI, FleurPlaza Europe, Grote-planten.nl, Gro4U, Greenport Logistics, Hamifleurs, Hamiplant, Holstein Flowers, Muscari Magic, NoSuchCompany, Nover Logistics, Noviflora, Oriental Group, OZ Import, Marjoland, Platform Agrologistiek, Rijnplant, Stet Heemskerk, TLN, Vida Verde, Free University of Amsterdam and Wageningen University.

Appendix A. Additional tables and figures

Table A.13
Summary of notations used in the mathematic formulation.

Indexes	
i, j	Index for hub locations $1, \ldots, \mathcal{N}$
v	Index for nodes $1, \ldots, \mathcal{V}$ in the mode-space-time network
$t, t', t_1, t_2, \tilde{t}, \hat{t}$	Index for time periods $1, \ldots, \mathcal{T}$
m'	Index for transportation mode $1, \ldots, \mathcal{M}$ in the real network
m, m_1, m_2	Index for mode $1, \ldots, \mathcal{M}+1$ in the mode-space-time network
(i, j)	Index representing a locational state (the travel arc between two locations if $i \neq j$, and the location arc if $i = j$)
(m_1, m_2)	Index representing a modal state (loading, unloading, if $m_1 \neq m_2$, and traveling, waiting, and holding, if $m_1 = m_2$)
p	Index for order $1, \ldots, \mathcal{P}$
Sets	
A	Set of all mode-space-time arcs $a_{(i,j),t,(m_1,m_2)}$
A_1	Set of all *travel arcs* in the mode-space-time network
A_2	Set of all *location arcs* in the mode-space-time network
$A_{t,m}$	Set of all arcs of transportation mode $m \in \{1, \ldots, \mathcal{M}\}$ passing time period t
Model Inputs (Parameters)	
T	Total number of time periods during the planning horizon (e.g. 48 h)
$\mathcal{P}$	Total number of orders
$\mathcal{M}$	Number of transportation mode types
S_i	Initial number of available empty RTIs at location i
$l_i^{(m_1,m_2)}, l_{i,t}^{(m_1,m_2)}$	Temperature of a modal state (m_1, m_2) inside location i at time t
$r_i, r_{i,t}^{(m_1,m_2)}$	Operation time inside location i starting at time t
C_i^{Hold}	Cost of holding RTIs at location i
$C_i^{(m_1,m_2)}$	General cost for handling and holding operations at location i
w_p	Demand (per number of RTIs) for order p
$O(p)$	Origin location for order p
$D(p)$	Destination location for order p
$PT(p)$	The time that order p is available for pickup at $O(p)$

(*continued on next page*)

Table A.13 (*continued*)

$DT(p)$	The latest delivery time for order p at $D(p)$
$\mathcal{L}_p$	Vaselife of order p ($time \times temperature$)
$\text{util}^{m'}$	Average utilization percentage of mode m'
VAT	A coefficient used to show the higher cost of transporting laden RTIs compared to empty ones
$F^{m'}$	Maximum number of vehicles for mode m'
$l^{m'}, l^{m'}_{(i,j),t}$	Temperature of mode m' for traveling between locations i and j starting at time t
$r^m, r^m_{(i,j),t}$	Travel time of mode m for traveling between locations i and j starting at time t
$speed^{m'}_{(i,j),t}$	Speed of mode m' for traveling between locations i and j starting at time t
$C^{m'}$	Fixed cost of operating mode m'
$C^{m'}_{\text{HR}}$	Cost of operating mode m' per hour
$C^{m'}_{\text{KM}}$	Cost of operating mode m' per Kilometer
$C^{m'}$	Cost of using one vehicle of mode m' ($C^{m'} = C^{m'}_{\text{fix}} + C^{m'}_{\text{KM}} \times dist_{(i,j)}$)
$C^{m'}_{\text{laden}}$	Variable cost of moving laden RTIs by mode m' per time period ($C^{m'}_{\text{laden}} = \text{VAT} \times C^{m'}_{\text{HR}}/(cap^{m'} \times \text{util}^{m'})$)
$C^{m'}_{\text{empty}}$	Variable cost of repositioning empty RTIs by mode m' per time period ($C^{m'}_{\text{empty}} = C^{m'}_{\text{HR}}/(cap^{m'} \times \text{util}^{m'})$)
$cap^{m'}$	Capacity of mode m'
Variables	
$\check{x}^{\text{laden}}_{p,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of laden RTIs for order p, on locational state (i, j) starting at time t, on modal state (m_1, m_2)
$\hat{x}^{\text{laden}}_{p,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of laden RTIs for order p, on locational state (i, j) finishing at time t, on modal state (m_1, m_2)
$\check{b}_{p,(i,j),t,(m_1,m_2)}$	Binary variable equal to 1, if flow of laden RTIs for order p is traversed on locational state (i, j) starting at time t, on modal state (m_1, m_2), and 0, if not
$\hat{b}_{p,(i,j),t,(m_1,m_2)}$	Binary variable equal to 1, if flow of laden RTIs for order p is traversed on locational state (i, j) finishing at time t, on modal state (m_1, m_2), and 0, if not
$\check{x}^{\text{assign}}_{p,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of assigned RTIs for order p, on locational state (i, j) starting at time t, on modal state (m_1, m_2)
$\hat{x}^{\text{assign}}_{p,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of assigned RTIs for order p, on locational state (i, j) finishing at time t, on modal state (m_1, m_2)
$\check{x}^{\text{repos}}_{p,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of repositioned RTIs for order p, on locational state (i, j) starting at time t, on modal state (m_1, m_2)
$\hat{x}^{\text{repos}}_{p,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of repositioned RTIs for order p, on locational state (i, j) finishing at time t, on modal state (m_1, m_2)
$\check{y}_{(i,j),t,(m_1,m_2)}$	Nonnegative variable representing the number of vehicles used on locational state (i, j) starting at time t, on modal state (m_1, m_2)
$\hat{y}_{(i,j),t,(m_1,m_2)}$	Nonnegative variable representing the number of vehicles used on locational state (i, j) finishing at time t, on modal state (m_1, m_2)
U^{laden}_{pit}	Real variable representing the demand (supply, if negative) of laden RTIs for order p at location i at time point t
U^{assign}_{pit}	Real variable representing the demand (supply, if negative) of assigned RTIs for order p at location i at time point t
U^{repos}_{pit}	Real variable representing the demand (supply, if negative) of repositioned RTIs for order p at location i at time point t
Additions for the Metaheuristic	
s	A solution
s_0	The initial solution
r_p, r_q	A route for order p or q
$f^m_{(i,j),t}$	A fleet of mode $m \in \{1, \ldots, \mathcal{M}\}$ used for the arc (i, j) at time t
q	Index for order $1, \ldots, \mathcal{Q}$
$\mathcal{Q}$	Total number of orders in each ALNS neighborhood
w_q	Demand (per number of RTIs) for order q
$O(q)$	Origin location for order q
$D(q)$	Destination location for order q
$PT(q)$	The time that order q is available for pickup at $O(q)$
$DT(q)$	The latest delivery time for order q at $D(q)$
$\mathcal{L}_q$	Vaselife of order q ($time \times temperature$)
C^m_{flow}	is equal to C^m_{laden} for laden flows, and C^m_{empty} for empty flows
$\check{x}^{\text{flow}}_{q,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of either laden or empty RTIs for order q, on location state (i, j) starting at time t, on mode state (m_1, m_2)
$\hat{x}^{\text{flow}}_{q,(i,j),t,(m_1,m_2)}$	Nonnegative variable representing flow of either laden or empty RTIs for order q, on location state (i, j) finishing at time t, on mode state (m_1, m_2)
$\check{b}_{q,(i,j),t,(m_1,m_2)}$	Binary variable equal to 1, if flow of laden RTIs for order q is traversed on location state (i, j) starting at time t, on mode state (m_1, m_2), and 0, if not
U^{flow}_{qit}	Real variable representing the demand (supply, if negative) of RTIs for order q at location i at time point t
$\check{x}^{\text{used}}_{p,(i,j),t,(m,m)}$	Parameter representing the flow of order p using the arc on location state (i, j) starting at time t, on mode state (m_1, m_2)
$\check{y}^{\text{used}}_{(i,j),t,(m_1,m_2)}$	Parameter representing the number of vehicles already operative on the arc on location state (i, j) starting at time t, on mode state (m_1, m_2)

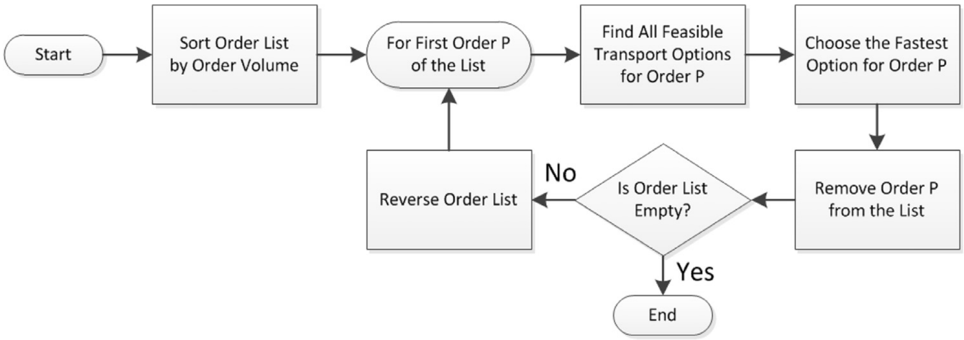


Fig. A.6. Bi-directional fastest fit algorithm.

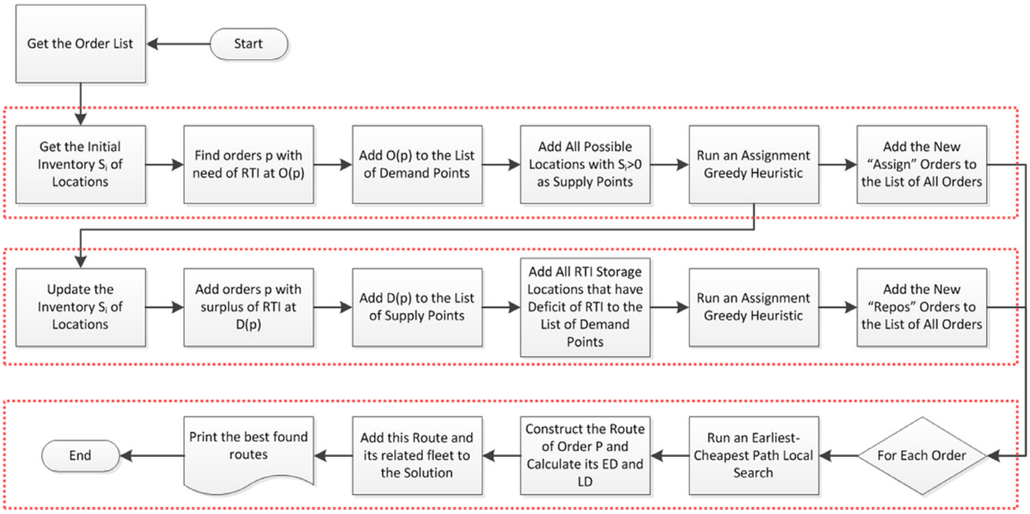


Fig. A.7. Initial solution generation.

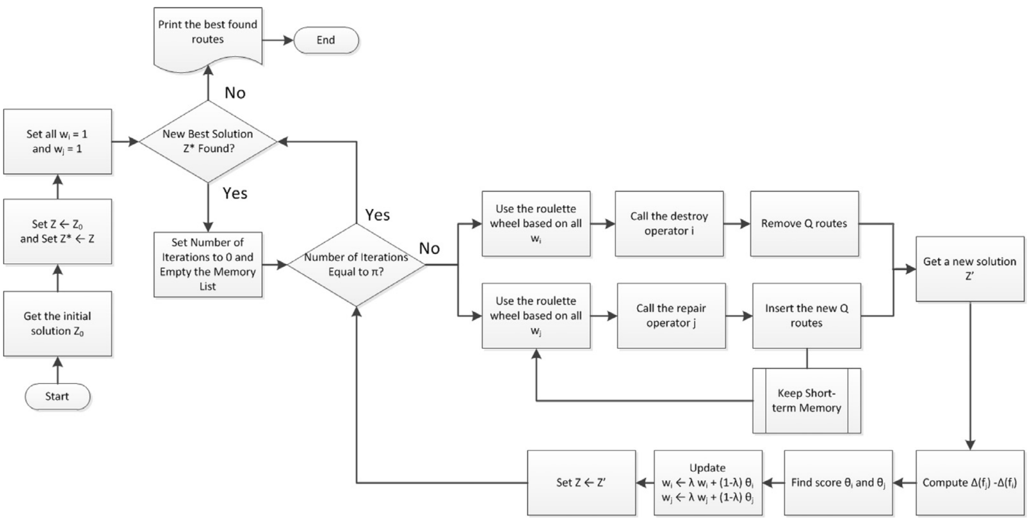


Fig. A.8. Adaptive large neighborhood search.

References

Agarwal, R., Ergun, O., 2008. Ship scheduling and network design for cargo routing in liner shipping. Transp. Sci. 42 (2), 175–196.
Ahumada, O., Villalobos, J., 2011. A tactical model for planning the production and distribution of fresh produce. Ann. Oper. Res. 190 (1), 339–358.

Andersen, J., Christiansen, M., Crainic, T., Grønhaug, R., 2011. Branch and price for service network design with asset management constraints. Transp. Sci. 45 (1), 33–49.
Andersen, J., Crainic, T., Christiansen, M., 2009a. Service network design with asset management: formulations and comparative analyses. Transp. Res. Part C 17 (2), 197–207.
Andersen, J., Crainic, T., Christiansen, M., 2009b. Service network design with management and coordination of multiple fleets. Eur. J. Oper. Res. 193 (2), 377–389.
Balakrishnan, A., Magnanti, T., Mirchandani, P., 1997. Network design. In: Dell'Amico, M., Maffoli, F., Martello, S. (Eds.), Annotated Bibliographies in Combinatorial Optimization, chapter 16, pp. 311–334.
Braekers, K., Caris, A., Janssens, G., 2013. Optimal shipping routes and vessel size for intermodal barge transport with empty container repositioning. Comput. Ind. 64 (2), 155–164.
Brouer, B., Desaulniers, G., Pisinger, D., 2014. A matheuristic for the liner shipping network design problem. Transp. Res. Part E 72, 42–59.
Brouer, B., Pisinger, D., Spoorendonk, S., 2011. Liner shipping cargo allocation with repositioning of empty containers. INFOR 49.
Canca, D., De-Los-Santos, A., Laporte, G., Mesa, J., 2017. An adaptive neighborhood search metaheuristic for the integrated railway rapid transit network design and line planning problem. Comput. Oper. Res. 78, 1–14.
Choong, S., Cole, M., Kutanoglu, E., 2002. Empty container management for intermodal transportation networks. Transp. Res. Part E 38 (6), 423–438.
Chouman, M., Crainic, T., 2010. A MIP-Tabu Search Hybrid Framework for Multicommodity Capacitated Fixed-Charge Network Design. Technical Report, 31. CIRRELT.
Chouman, M., Crainic, T., 2014. Cutting-plane matheuristic for service network design with design-balanced requirements. Transp. Sci. 49 (1), 99–113.
Chouman, M., Crainic, T., Gendron, B., 2011. Commodity representations and cutset-based inequalities for multicommodity capacitated fixed-charge network design. Technical Report, 56. CIRRELT.
Costa, A., 2005. A survey on benders decomposition applied to fixed-charge network design problems. Comput. Oper. Res. 32 (6), 1429–1450.
Costa, A., Cordeau, J., Gendron, B., 2009. Benders, metric and cutset inequalities for multicommodity capacitated network design. Comput. Optim. Appl. 42 (3), 371–392.
Crainic, T., Hewitt, M., Toulouse, M., Vu, D., 2016. Service network design with resource constraints. Transp. Sci. 50 (4), 1380–1393.
Crainic, T., Li, Y., Toulouse, M., 2006. A first multilevel cooperative algorithm for capacitated multicommodity network design. Comput. Oper. Res. 33 (9), 2602–2622.
Demir, E., Burgholzer, W., Hrušovský, M., Arıkan, E., Jammernegg, W., van Woensel, T., 2015. A green intermodal service network design problem with travel time uncertainty. Transp. Res. Part B 1–19.
Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., Vogel, U., 2011. Vehicle routing with compartments: applications, modelling and heuristics. OR Spectr. 33 (4), 885–914.
Doerner, K., Gronalt, M., Hartl, R., Kiechle, G., Reimann, M., 2008. Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows. Comput. Oper. Res. 35 (9), 3034–3048.
Erera, A., Morales, J., Savelsbergh, M., 2005. Global intermodal tank container management for the chemical industry. Transp. Res. Part E 41 (6), 551–566.
FloraHolland, 2015. Annual report.
García, J., Florez, J., Torralba, A., Borrajo, D., López, C., García-Olaya, A., Sáenz, J., 2013. Combining linear programming and automated planning to solve intermodal transportation problems. Eur. J. Oper. Res. 227 (1), 216–226.
Gendron, B., Larose, M., 2014. Branch-and-price-and-cut for large-scale multicommodity capacitated fixed-charge network design. EURO J. Comput. Optim. 2 (1–2), 55–75.
Ghamlouche, I., Crainic, T.G., Gendreau, M., 2003. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. Oper. Res. 51 (4), 655–667.
Gharehgozli, A., Laporte, G., Yu, Y., de Koster, R., 2014. Scheduling twin yard cranes in a container block. Transp. Sci. 49 (3), 686–705.
Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. Eur. J. Oper. Res. 254 (1), 80–91.
Hemmati, A., Hvattum, L., Christiansen, M., Laporte, G., 2016. An iterative two-phase hybrid matheuristic for a multi-product short sea inventory-routing problem. Eur. J. Oper. Res. 252 (3), 775–788.
Hewitt, M., Nemhauser, G., Savelsbergh, M., 2010. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. INFORMS J. Comput. 22 (2), 314–325.
Hewitt, M., Nemhauser, G., Savelsbergh, M., 2013. Branch-and-price guided search for integer programs with an application to the multicommodity fixed-charge network flow problem. INFORMS J. Comput. 25 (2), 302–316.
Hsu, C., Hung, S., Li, H., 2007. Vehicle routing problem with time-windows for perishable food delivery. J. Food Eng. 80 (2), 465–475.
Karsten, C., Brouer, B., Desaulniers, G., Pisinger, D., 2016. Time constrained liner shipping network design. Transp. Res. Part E. Article in press.
Li, X., Wei, K., Aneja, Y., Tian, P., 2016. Design-balanced capacitated multicommodity network design with heterogeneous assets. Omega. Article in press.
Mauri, G., Ribeiro, G., Lorena, L., Laporte, G., 2016. An adaptive large neighborhood search for the discrete and continuous berth allocation problem. Comput. Oper. Res. 70, 140–154.
Mendoza, J., Castanier, B., Guéret, C., Madaglia, A., Velasco, N., 2011. Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. Transp. Sci. 45 (3), 346–363.
Meng, Q., Wang, S., 2011. Liner shipping service network design with empty container repositioning. Transp. Res. Part E 47 (5), 695–708.
Moccia, L., Cordeau, J., Laporte, G., Ropke, S., Valentini, M., 2011. Modeling and solving a multimodal transportation problem with flexible time and scheduled services. Networks 57 (1), 53–68.
Osvald, A., Stirn, L., 2008. A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. J. Food Eng. 85 (2), 285–295.
Paraskevopoulos, D., Bektaş, T., T.G., C., Potts, C., 2016. A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem. Eur. J. Oper. Res. 253 (2), 265–279.
Pedersen, M., Crainic, T., Madsen, O., 2009. Models and tabu search metaheuristics for service network design with asset-balance requirements. Transp. Sci. 43 (2), 158–177.
Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Comput. Oper. Res. 34 (8), 2403–2435.
Reis, S., Leal, J., 2015. A deterministic mathematical model to support temporal and spatial decisions of the soybean supply chain. J. Transp. Geogr. 43, 48–58.
Rifai, A., Nguyen, H., Dawal, S., 2016. Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling. Appl. Soft Comput. 40, 42–57.
Rodríguez-Martín, I., Salazar-González, J., 2010. A local branching heuristic for the capacitated fixed-charge network design problem. Comput. Oper. Res. 37 (3), 575–581.
Ropke, S., Pisinger, D., 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.
Rosenboom, A., 2014. Development of a decision support system for an orchestrator in the horticultural sector. TUe School of Industrial Engineering Master Thesis.
Sloof, M., Everest, B., 2001. Problem decomposition. In: Tijskens, L., Hertog, M., Nicolai, B. (Eds.), Food Process Modeling,, pp. 19–34. chapter 2
Song, D., Dong, J., 2012. Cargo routing and empty container repositioning in multiple shipping service routes. Transp. Res. Part B 46 (10), 1556–1575.
SteadieSeifi, M., Dellaert, N., Nuijten, W., van Woensel, T., Raoufi, R., 2014. Multimodal freight transportation planning: a literature review. Eur. J. Oper. Res. 233 (1), 1–15.

Taoukis, P., 2001. Modeling the use of time-temperature indications in distribution and stock rotations. In: Tijskens, L., Hertog, M., Nicolai, B. (Eds.), Food Process Modeling, pp. 19–34. chapter 19
Tarantilis, C., Kiranoudis, C., 2001. A meta-heuristic algorithm for the efficient distribution of perishable food. J. Food Eng. 50 (1), 1–9.
Tarantilis, C., Kiranoudis, C., 2002. Distribution of fresh meat. J. Food Eng. 51 (1), 85–91.
Thiongane, B., Cordeau, J., Gendron, B., 2015. Formulations for the nonbifurcated hop-constrained multicommodity capacitated fixed-charge network design problem. Comput. Oper. Res. 53, 1–8.
Tosi, M., 2014. Allocation of VALS activities in a metro model network in the European floricultural market. Wageningen Department of Social Sciences Master Thesis,.
Tromp, S., van der Sman, R., Vollebregt, H., Woltering, E., 2012. On the prediction of the remaining vase life of cut roses. Postharvest Biol. Technol. 70, 42–50.
Verhoeven, S., 2014. Hub location decisions in the metromodel. TUe School of Industrial Engineering Master Thesis.
Vlassak, T., 2014. Future-proof return logistics in the floriculture sector. TUe School of Industrial Engineering Master Thesis.
Yaghini, M., Karimi, M., Rahbar, M., Sharifitabar, M., 2014. A cutting-plane neighborhood structure for fixed-charge capacitated multicommodity network design problem. INFORMS J. Comput. 27 (1), 48–58.
Zhu, E., Crainic, T., Gendreau, M., 2014. Scheduled service network design for freight rail transportation. Oper. Res. 62 (2), 383–400.