**ORIGINAL RESEARCH**

# A learning-based co-planning method with truck and container routing for improved barge departure times

**Rie B. Larsen[1]** · **Rudy R. Negenborn[1]** · **Bilge Atasoy[1]**

## Abstract

Cooperation between container transport service providers can increase efficiency in the logistics sector significantly. However, cooperation between competitors requires co-planning methods that not only give the cooperating partners an advantage towards external competition but also protect the partners from losing information, clients and autonomy to one another. Furthermore, modern freight transport requires real-time methods that react to new information and situations. We propose a real-time, co-planning method called *departure learning* based on model predictive control where a barge operator considers the joint cost of themselves and a truck operator when deciding barge departures. At regular time-intervals, the barge operator uses previous information to propose a number of departure schedules for which the truck operator discloses their corresponding expected operational costs. Co-planning thus only requires limited exchange of aggregate data. The impact of using departure learning on the transport system's performance and the method's learning quality are thoroughly investigated numerically on an illustrative, simulated, realistic hinterland network. With as little as six schedules being exchanged per timestep, departure learning outperforms decentralized benchmark methods significantly in terms of operational costs. It is found that using knowledge about the performance of related schedules is important for the exploration of opportunities, but if this is relied upon too much, the realized solution becomes more costly. It is also found that departure learning is a reliable and realistic co-planning method that especially performs well when peaks in the demand make departure times highly correlated to the cost of operating the transport system, such as in hinterland areas of ports which receive large container ships.

✉ Rie B. Larsen
  R.B.Larsen@tudelft.nl

  Rudy R. Negenborn
  R.R.Negenborn@tudelft.nl

  Bilge Atasoy
  B.Atasoy@tudelft.nl

[1] Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

## 1 Introduction

Better *co-planning* between stakeholders in transport systems for planning barge schedules, truck and container routes in real-time will help utilizing the transport capacity better. A more efficient transport system will help alleviating the negative impacts on the environment, since less resources will be used and less pollution will be emitted on each transport. The transport sector is a large contributor of $CO_2$ emissions and has a low efficiency, with, e.g., trucks being empty 26% of the kilometres they drive in the Netherlands (Eurostat, 2020). $CO_2$ emission is however not the only negative impact of freight transport. The report van Essen et al. (2019) estimates the external costs of transport, such as the cost of accidents, climate impact, and noise nuisance. Here it is concluded that maritime transport induces the lowest external cost, followed by rail, inland waterway and road transport in this order. It is therefore desirable not only to improve the vehicle utilization, and hence efficiency, of truck transport, but also the utilization across transport modes.

Synchromodal transport uses a-modal bookings and change acceptance to enable transport providers to optimize plans in accordance with the realisation of uncertainties (Giusti et al., 2019; van Riessen et al., 2015). The concept gives transport providers more flexibility than the previous concepts, intermodal and multimodal transport, since mode decisions in these concepts were fixed, at latest, before departure. In the traditional transport literature, decisions are divided into strategic, tactical and operational levels (SteadieSeifi et al., 2014). Strategic decisions have long lasting impact and usually high impact on revenue. Tactical decisions have impact over a tangible time horizons and are typically based on estimates of future events. Plans are often made on the tactical level and corrected at the operational level. Operational decisions regard what to do right now with the realised events. With synchromodal transport, decisions from the tactical and the operational levels are intertwined: uncertain long term plans for operational decisions can be formulated without commitment, and tactical decisions can be changed during operation. This intertwining requires additional research to utilize the potential of synchromodality. Model Predictive Control (MPC) provides a framework for combining predictions of future events with real-time decision making. MPC has previously been used to route containers in several cases, e.g., Huang et al. (2021), and Li et al. (2015).

Barge schedules are typically decided on at the tactical level based on estimated demand (Demir et al., 2016). When plans are made in advance, the realised demand is often different and external factors, like weather, cause unforeseen limitations. Some methods plan in accordance with these uncertainties (van Riessen et al., 2015), others adjust predefined departure times after the demand realization (Behdani et al., 2016) or cancel unprofitable departures (Xu et al., 2015). Truck routing is typically decided on at the operational level based on pick up and delivery locations and times of the goods (Psaraftis et al., 2016). In Larsen et al. (2021c) we demonstrated the negative impact of planning first container routes and then truck routes compared to planning them simultaneously in a synchromodal network. The results of Qu et al. (2019) show the same on a network with only one origin of the demand.

Barges and trucks are often operated by different stakeholders, so simultaneous planning requires cooperation. Cooperation can involve both information sharing and loss of autonomy. Many companies are interested in the benefits of cooperation (Cruijssen et al., 2007), but participate reluctantly due to these implications. In Karam et al. (2021), it is concluded that

trust-related issues are the main barriers to horizontal cooperation. Information on specific transport requests are often considered sensitive information. In Gansterer et al. (2020), it is shown that sharing of aggregated information can improve the system-wide performance in comparison with sharing of more sensitive, specific information. Many cooperation schemes in the transport literature are constructed such that missing information or sudden changes in the willingness to follow the scheme can damage the other participating parties. Traditional cooperation schemes vary from auctions (Xu et al., 2017) to distributed optimization (Li et al., 2017; Di Febbraro et al., 2016). Gansterer and Hartl (2018) provides an overview that classifies the existing research into (1) centralized methods, which requires a neutral party, (2) decentralized optimization methods, and (3) auction-based methods.

We use the term *co-planning* to describe the act of cooperating to achieve the vehicle and container transport plans that are best for the group of cooperating stakeholders without sharing sensitive information or being vulnerable to defiance of the other parties. Very little research on co-planning methods exist. Boros et al. (2008) proposes a co-planning method that optimizes the joint profit cycle-time imposes on a port and a shipping liner. In their method in each round of communication, each operator optimizes its costs for a given cycle-time. These costs are communicated and the expected joint profit is computed. If the expected profit has improved, a higher cycle-time is used in the next round of communication. Since the joint problem under certain conditions is concave, the joint profit and the corresponding cycle-time can be optimized without exchanging detailed information. For more complex problems, their method may find the lowest cycle-time corresponding to a local maxima, not the global maximum. The planning method presented in Pérez Rivera and Mes (2019), which integrates drayage and long-haul parts of container routing, separates the optimization of each type of operation but ensures a common goal by using a shared objective function and coordinate demand-expectation information across the models. In the literature on transport contract negotiation, planning at a tactical level is researched under privacy assumptions similar to those of co-planning. In Yang et al. (2019) a carrier offers two contracts to a shipper. Since it is a tactical problem, the carrier only has probabilistic information about the future demand when deciding what contracts to propose. For operational co-planning, the decision frequency is higher, and there is thus a potential to learn from the other stakeholders' previous actions.

In Bayesian optimization, prior knowledge is used to identify function evaluations that are likely to improve the optimizer's knowledge of the minimum by means of a value function that typically combines certainty with expected function values. It is often used to optimize functions where gradient information is unavailable or computationally expensive and where function evaluations are associated with a cost. Each new function evaluation adds to the prior knowledge (Frazier, 2018). Bayesian optimization has earlier been used for container routing problems in Rivera and Mes (2022), where a value function is used to improve the trade-off between exploration and exploitation in an Approximate Dynamic Programming solution method. Ideas from Bayesian optimization is used in the present paper to decrease the needed communication between the barge and the truck operator. This is a novel application in the transport field. The presented method is inspired by the literature on combinatorial Bayesian optimization (Oh et al., 2019; Baptista and Poloczek, 2018), parallel Bayesian optimization (Očenášek and Schwarz, 2000; Kandasamy et al., 2018), and warm-start of Bayesian optimization (Poloczek et al., 2016).

In this paper, we show the impact of co-planning and describe the details of a co-planning method, called *departure learning*, for real-time co-planning between a barge and a truck operator. The co-planned actions are departures of a barge and the remaining actions considered are loading and unloading of containers to trucks and barge, and departures of the trucks.

An initial version of departure learning was presented in Larsen et al. (2020) together with preliminary results and a variation of this method for one barge and multiple truck operators was presented in Larsen et al. (2021b), again with limited exploration of the method's features and limited experimental work. The method is based on Model Predictive Control (MPC) and uses ideas from Bayesian optimization to learn good departure times through continuous communication. Departure learning requires communication of a number of schedules and indications of the corresponding performances between the barge operator and the truck operator. The initial departure learning algorithm is now enhanced with better initial guesses for the performance of schedules and the impact of the method's learning parameters is presented. Furthermore, we show the impact of actively learning what schedules are expected to perform well by comparing departure learning to a similar method that uses randomly chosen schedules. It is assumed that no party seeks to exploit the framework, but if one party acts autonomously the other party is not damaged. The main advantage of departure learning is the ability to co-plan barge departures, i.e. enable a barge operator to depart when it improves the operation cost of the transport network, without losing control over own operation, transferring responsibilities between stakeholders or communicating detailed information.

This paper is organized as follows. Section 2 details the problem of co-planning barge departures. In Sect. 2.1, an MPC planning barge, truck and container moved in a network with a single decision maker is presented. Hereafter, departure learning is introduced in Sect. 3. The overall framework is presented first, followed by the barge operator's learning strategy and the truck operator's feedback. In Sect. 4, simulation experiments are used to show the impact of the parameters of departure learning and how departure learning performs compared to three benchmark methods. Section 5 concludes the paper and provides future research directions.

## 2 Problem statement

When each transport operator can change planned actions up until the time the action is carried out, co-planning between transport operators must happen in real time. One planning problem in a synchromodal transport network is co-planning between truck operators responsible for routing trucks and delivering containers in time, and a barge operator responsible for barge departures. Figure 1 shows the synchromodal transport network used as example in this paper. The simple road network is used for illustrative purposes. In reality road networks usually have more nodes corresponding to locations where decisions can be taken. The method is described for arbitrary road network size with one truck operator and extends easily to multiple truck operators as described in Larsen et al. (2021b) without a thorough investigation of the impact of the method's parameters on the achieved performance.

Changing the departure time of a barge impacts the truck operator in the network significantly. If the barge departure fits well with the release and due date of container demand, it is often attractive for the truck operator to send the containers by barge instead of trucking them. They can do so as the containers are booked a-modal under synchromodal transport. Good departure times thus often benefit the truck operator by decreasing transport cost and the barge operator by increasing the transported volume. The barge and the truck operator thus both have an interest in co-planning to achieve a better joint operational cost. We assume the barge and truck operators have an agreement outlining the distribution of the economical gains and burdens. However, the two operators do not want to share specific information nor are they willing to hand over decision power.

**Fig. 1** Small Dutch transport network used as example in this paper
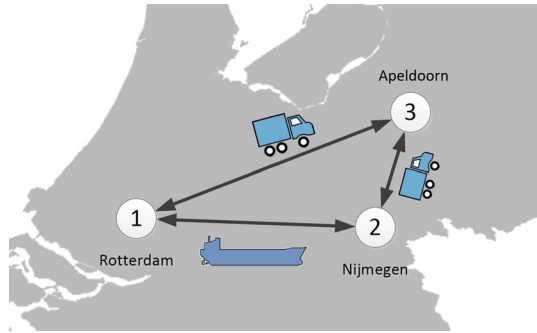


The barge operator does not know which barge departure times will decrease the joint cost of the transport the most. In a highly dynamic environment with many small shippers that do not request transport of containers regularly, optimizing barge departures based on past transport flows can give very ill-fitting schedules for upcoming periods. In those cases, assuming the barge operator has no knowledge about demand but what he receives in real-time from the truck operators is thus reasonable. However, the truck operator wants to keep all information about a given container's transport-order (such as release time, due date and quantity) private until the container is committed to take the barge. Only when a container is committed to be loaded on the barge, the truck operator is willing to share the necessary practical information. The truck operator is willing to share the total cost they expect to occur over a certain time horizon if a given barge schedule is implemented. They do however prefer to limit the number of schedules to respond to, as computing a plan for each schedule can become computationally demanding and they worry the barge operator will gain too much insight into their business economy.

On the other hand, the truck operator do not have authority to decide on the barge schedule as it is a core business decision for a barge operator. Additionally, in realistic cases, multiple stakeholders will use the capacity on the barge ensuring no single party has full overview of what departure times will be most beneficial. For simplicity we assume a single truck operator. We consider a transport system that consists of a barge operator who decides on the departure times of one barge between two different terminals and one other operator, namely the truck operator, who decides on the routing of trucks and the mode and route containers are transported by. We, furthermore, assume in the numerical experiments that the truck operator shares the cost truly and that the barge operator indeed sends back the best schedule honestly. It is, however, worth noticing that each operator can incorporate other expenses into the costs used in the method and has full authority over actions they are responsible for.

If one entity had the joint information and decision power from the barge and the truck operator, full integration would be possible. Full integration, typically, enables the best possible performance of the system, but is unattainable when multiple stakeholders have reservations towards sharing information and responsibilities. In short the presented method, called *departure learning*, answers the question 'How can a barge operator learn which barge departures make the total operation cost of a synchromodal transport network lower under limited real-time information about the total cost?'. To clarify assumptions, model choice and to provide an idealized solution method, we introduce first a real-time, centralized method. If one entity has full knowledge and authority in the transport system, this method provides receding horizon-optimal control of the integrated container, truck and vehicle departures.In

Sect. 3, we propose *departure learning* to let the performance of the decentralized transport system approach that of the centralized method.

## 2.1 Real-time centralized decision making

To fully integrate the barge departure scheduling with container and truck routing, a model predictive control (MPC) based method has been developed. Except the assumed central decision agent, the assumptions and notation of this centralized method are also applicable to departure learning, e.g., the underlying transport network dynamics are identical. A comprehensive table of notation can be found in Appendix A.

MPC addresses uncertainties by adjusting future plans based on feedback from the system (Mayne et al., 2000). Every $\Delta t$ timeunits there is a new decision moment where actions to be taken until the next decision moment are fixed based on a predicted plan for the next $T_p \Delta t$ timeunits. We denote with $t$ the running time and with $k$ the count of decision moments. We call the latter timesteps and define $t = k \Delta t$.

Using MPC for problems that require frequent updates, i.e. low $\Delta t$, and a long prediction horizon $T_p$ necessitates fast optimization of the model. In synchromodal transport problems, a long prediction horizon is needed because of the long travel times of barges and the need to describe at least one departure from each terminal. We therefore assume local controllers can aggregate the present containers and trucks into commodities based on their destination. Additionally, each commodity is described using continuous variables. This decreases the computational complexity sufficiently such that no heuristics, such as the ones described in Juan et al. (2021), are needed. Frequent updates can ensure sufficient precision when the continuous optimal decisions are rounded to integer variables (Sager et al., 2012). We separate the containers into commodity flows based on their destinations. If a finer granularity is needed, due date and container type can also be included in the definition of a commodity. This has earlier been considered in, e.g., Nabais et al. (2013), which differentiates between containers with different destination and due time combinations, and Poo and Yip (2019), which only distinguishes full and empty containers. A finer granularity will increase the computational complexity and thus the time period, $\Delta t$, between decision moments. It is assumed local decision makers disaggregate the commodity flow decisions into actions for separate containers. The barge capacity is much larger than that of trucks and the vessel movements are thus described by binary variables.

The key assumptions of the synchromodal transport system dynamics are:

- Any node in the network can be the origin and destination of transport demand, if it is defined as such, hence bidirectional flows are considered.
- Demand is modelled as containers available to the network and needed from the network. Unsatisfied demand is penalized. The demand is fully known over the prediction horizon.
- Containers are modelled as flows segmented by their destination.
- Trucks are modelled as flows.
- The number of trucks is finite and each truck can transport one container.
- The barge has invariant, finite capacity.
- Quay capacity, crane capacity, etc., and (un)loading rates are considered sufficient.
- Terminal operating hours, drivers resting hours, etc., are not considered.

Each geographical location in the synchromodal transport network is represented by a node in a graph with multiple directed arcs connecting the nodes. A location can be a terminal where trucks and/or the barge can be loaded and unloaded; a way point where trucks can change their intended route; or a hub where trucks can pick up or deliver containers and

possibly park. The set of nodes is denoted $\mathcal{N}$ and the set of road-arcs is denoted $\mathcal{R}$. There is one barge in the network that sails between node 1 and node 2. The two directional arcs describing this waterway comprise the set $\mathcal{W}$. Nodes 1 and 2 can also be connected by road. The road network can be of any size, but must contain at least two locations that function as origin and destination of containers. The operators' decisions can only be changed when the vehicles and containers are at the nodes. It is, e.g., not possible to make the barge return to its departure terminal if a delay occurs. Furthermore, it is assumed that only the truck operators have contact to clients and therefore the barge operator receives the demand only through truck operators.

## 2.2 System dynamics

In the following we describe the realized dynamics of the transport system. Departure learning and the centralized method are both based on predictions of the consequences of future actions. These predictions are made using the same dynamics. To distinguish between predicted and realized actions and states, we use a bar over the notation for the realized case. The method used to describe this simultaneous truck and container planning problem is a simplification of the one presented in Larsen et al. (2021a). The following assumptions allow the use of a simplified method: 1) the travel time is deterministic, 2) all trucks are of the same kind, 3) there is no scheduled services in the transport network, and 4) there is unlimited capacity for loading and unloading. The full method can be used with departure learning, but as it adds complexity to the description, the simplified model is used here.

We define virtual demand nodes adjacent to the graph nodes where containers can have origin or destination. They act as a reminder of unfulfilled bookings. The set of virtual nodes is denoted by $\mathcal{D}$. The dynamics of the virtual demand nodes are given as

$$\bar{z}_i^d(k+1) = \bar{z}_i^d(k) - \bar{u}_{di}(k) - \bar{u}_{id}(k) + \bar{d}_i(k) \quad \forall i \in \mathcal{D}, \ \forall k, \tag{1}$$

where $\bar{d}_i(k) \in \mathbb{R}_{\geq 0}^{n_c}$ is the newly realized demand of each commodity at time $k$. Notice that all values are positive, so whether the demand indicates containers that are ready to be transported or that are due depends on the commodity, i.e. the element in the vector. $n_c$ is the number of commodities, i.e. the number of virtual demand nodes. The mappings $p_i^r \in \{0, 1\}^{1 \times n_c}$ and $p_i^d \in \{0, 1\}^{1 \times n_c}$ are defined such that $p_i^r \bar{d}_i(k)$ is the sum of containers that are ready to be picked up at node $i$ at time $k$ and $p_i^d \bar{d}_i(k)$ is the sum of containers that are due at node $i$ at time $k$. The variable $\bar{z}_i^d(k) \in \mathbb{R}_{\geq 0}^{n_c}$ is the unsatisfied demand at node $i$ at time $k$ and $\bar{u}_{id}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ is the containers of each commodity from terminal node $i$ that are used to satisfy due dates at the virtual demand node at time $k$. $\bar{u}_{di}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ is the opposite. To guide the direction of the demand satisfaction, the following must be true:

$$p_i^r \bar{u}_{id}(k) = 0 \quad \forall i \in \mathcal{D}, \ \forall k \tag{2}$$
$$p_i^d \bar{u}_{di}(k) = 0 \quad \forall i \in \mathcal{D}, \ \forall k \tag{3}$$

Each node in the network can be connected with three kinds of other nodes: $\mathcal{D}_i$, $\mathcal{W}_i$ and $\mathcal{R}_i$. $\mathcal{D}_i$ contains node $i$'s adjacent virtual demand nodes and $\mathcal{W}_i$ the node to which $i$ is connected by waterways. These sets are either empty or have one element. The set $\mathcal{R}_i$ contains all nodes that are connected to node $i$ by road. Based on these sets, the dynamics of the stacks of containers are

$$\bar{z}_i^c(k+1) = \bar{z}_i^c(k) + \sum_{j \in \mathcal{D}_i} (\bar{u}_{di}(k) - \bar{u}_{id}(k)) + \sum_{j \in \mathcal{W}_i} \left( \bar{u}_{ji}^b(k - \tau_{ji}^b) - \bar{u}_{ij}^b(k) \right)$$

$$+ \sum_{j \in \mathcal{R}_i} \left( \bar{u}_{ji}(k - \tau_{ji}^r) - \bar{u}_{ij}(k) \right) \quad \forall i \in \mathcal{N}, \forall k. \tag{4}$$

The variable $\bar{z}_i^c(k) \in \mathbb{R}_{\geq 0}^{n_c}$ is a vector of how many containers of each commodity that are stacked at node $i$ at time $k$. $u_{ij}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ has the same structure and is for the containers transported from $i$ to $j$ by road at time $k$. The road travel between $i$ and $j$ takes $\tau_{ij}^r$ timesteps and the waterway travel takes $\tau_{ij}^b$. $u_{ij}^b(k) \in \mathbb{R}_{\geq 0}^{n_c}$ is a vector with the number of containers of each commodity that is transported from $i$ to $j$ by barge departing at time $k$.

Two binary variables $y_1(k)$ and $y_2(k)$ are used to describe the departures of the barge at timestep $k$ from node 1 and 2 respectively. The travel time from node 1 to 2, $\tau_{12}^b$, and the return, $\tau_{21}^b$, include loading, travel time, mooring and unloading. Containers that arrive at the terminal after loading has started will not be accepted on the barge and containers can only be picked up after the barge has finished unloading all containers. The dynamics of the barges is described as

$$\bar{z}_i^b(k+1) = \bar{z}_i^b(k) - \bar{y}_i(k) + \bar{y}_j(k - \tau_{ji}^b) \quad i, j \in \{1, 2\}, i \neq j, \forall k \tag{5}$$

where $\bar{z}_i^b(k) \in \{0, 1\}$ is the number of barges at the quay of node $i$ at time $k$.

The barge has a capacity of $c^b$ and only carries containers that were ready for loading at the departure time. Hence

$$\mathbf{1}_{n_c} \bar{u}_{ij}^b(k) \leq c^b \bar{y}_i(k) \quad \forall <i, j> \in \mathcal{W}, \forall k, \tag{6}$$

where $\mathbf{1}_{n_c} \in \mathbb{R}^{1 \times n_c}$ is a vector of ones. The variable $\bar{z}_i^v(k) \in \mathbb{R}$ is the number of trucks parked at node $i$ at time $k$, and has the dynamics

$$\bar{z}_i^v(k+1) = \bar{z}_i^v(k) + \sum_{j \in \mathcal{R}_i} \bar{v}_{ji}(k - \tau_{ji}^r) - \bar{v}_{ij}(k) \quad \forall i \in \mathcal{N}, \forall k, \tag{7}$$

where $\bar{v}_{ij}(k) \in \mathbb{R}$ is the number of trucks departing from $i$ on the road to $j$ at time $k$. To ensure containers only travel by roads if they are loaded on trucks, the sum of containers departing node $i$ at time $k$ on the road to node $j$ must not exceed the number of trucks departing on the same road at the same time. Trucks are on the other hand allowed to drive empty. Both are modelled by

$$\mathbf{1}_{n_c} \bar{u}_{ij}(k) \leq \bar{v}_{ij}(k) \quad \forall j \in \mathcal{R}_i, \forall i \in \mathcal{N}, \forall k. \tag{8}$$

## 2.3 Centralized method

At each discrete timestep, the centralized method optimizes the predicted cost of operating the synchromodal transport system over the time horizon $k$ to $k + T_p$. It is assumed that it is cheaper but slower to use the barge than to only use the road mode if we look isolated at sending one container between the waterway terminals on a barge with a realistic utilization and do not consider the cost of driving a truck empty. The actual cost will depend on the vehicles' utilization and the container's origin and destination. We consider four kind of operation costs:

$w_i^b \in \mathbb{R}$      Total operation cost associated with an empty barge that departs from node $i$

$w_{ij}^l \in \mathbb{R}_{\geq 0}^{1 \times n_c}$      Cost of transporting one additional container with the barge from $i$ to $j$. Can vary based on commodity

$w_{ij}^v \in \mathbb{R}$      Cost of driving a truck from $i$ to $j$, regardless of load

$w_d \in \mathbb{R}_{\geq 0}^{1 \times n_c}$    Cost per timestep delay per container. Can vary based on commodity. Demand satisfaction is formulated as a soft constraint

The base cost of sailing the barge is defined as:

$$J^b(k) = w_1^b y_1(k) + w_2^b y_2(k) \quad \forall k. \tag{9}$$

It is assigned to the timestep where the barge departs, i.e. the total travel-cost is incurred at departure and not during the travel. This reflects the assumption that plans for a specific vehicle can only be changed when that vehicle is at a node. Running costs like owning the equipment and hiring people are disregarded, as they are out of scope of the real-time, operational problem.

The remaining cost of operating the synchromodal network is, as the barge cost, assigned to the timestep a truck or a container departs. The soft constraint penalty for late delivery of containers at their destinations is added per container, per timestep. The remaining cost is thus:

$$J^t(k) = \sum_{<i,j> \in \mathcal{R}} w_{ij}^v v_{ij}(k) + \sum_{<i,j> \in \mathcal{W}} w_{ij}^l u_{ij}^b(k) + \sum_{i \in \mathcal{D}} w_d z_i^d(k+1), \tag{10}$$

At each timestep, the centralized method gathers the number of containers, trucks and barge that are located at each node and the quantities due to arrive in the future as a consequence of previous decisions. The information related to the barge location is:

$$\bar{x}^b(k) = \left[ \bar{z}_1^b(k), \, \bar{y}_2(k-1), \, \cdots, \, \bar{y}_2(k-\tau_{12}^b), \, \bar{z}_2^b(k), \, \bar{y}_1(k-1), \, \cdots, \, \bar{y}_1(k-\tau_{21}^b) \right]^T \tag{11}$$

and the remaining information is:

$$\bar{x}^t(k) = \begin{bmatrix} \left[ \bar{z}_1^d(k) \cdots \bar{z}_{|\mathcal{D}|}^d(k) \right]^T \\ \left[ \bar{u}_{j1}^b(k-\tau_{j1}^b), j \in \mathcal{W}_1, \cdots, \bar{u}_{j|\mathcal{N}|}^b(k-\tau_{j|\mathcal{N}|}^b), j \in \mathcal{W}_{|\mathcal{N}|} \right]^T \\ \left[ \bar{z}_1^c(k), \bar{u}_{j1}(k-\tau_{j1}^r) \, \forall j \in \mathcal{R}_1, \cdots, \bar{z}_{|\mathcal{N}|}^c(k), \bar{u}_{j|\mathcal{N}|}(k-\tau_{j|\mathcal{N}|}^r) \, \forall j \in \mathcal{R}_{|\mathcal{N}|} \right]^T \\ \left[ \bar{z}_1^v(k), \bar{v}_{j1}(k-\tau_{j1}^r) \, \forall j \in \mathcal{R}_1, \cdots, \bar{z}_{|\mathcal{N}|}^v(k), \bar{v}_{j|\mathcal{N}|}(k-\tau_{j|\mathcal{N}|}^r) \, \forall j \in \mathcal{R}_{|\mathcal{N}|} \right]^T \end{bmatrix}. \tag{12}$$

This information forms the initial constraints of optimization problem (13)–(24), which is used to decide what actions to implement at the current timestep $k$. At the next timestep, the process is repeated such that at any timestep $k$ the actions and their consequences are optimized for $k$ to $k + T_p$, but only the actions corresponding to $k$ are implemented.

$$\min \sum_{\kappa=k}^{k+T_p-1} J^t(\kappa) + J^b(\kappa) \tag{13}$$

$$\text{s.t. } x^t(k) = \bar{x}^t(k), \tag{14}$$

$$x^b(k) = \bar{x}^b(k), \tag{15}$$

$$z_i^d(\kappa+1) = z_i^d(\kappa) - u_{di}(\kappa) - u_{id}(\kappa) + d_i(\kappa) \quad \forall i \in \mathcal{D}, \, \forall \kappa \in \{k, \ldots, k+T_p-1\}, \tag{16}$$

$$z_i^c(\kappa+1) = z_i^c(\kappa) + \sum_{j \in \mathcal{D}_i} (u_{di}(\kappa) - u_{id}(\kappa)) + \sum_{j \in \mathcal{W}_i} \left( u_{ji}^b(\kappa - \tau_{ji}^b) - u_{ij}^b(\kappa) \right)$$

$$+ \sum_{j \in \mathcal{R}_i} \left( u_{ji}(\kappa - \tau_{ji}^r) - u_{ij}(\kappa) \right) \quad \forall i \in \mathcal{N}, \forall \kappa \in \{k, \ldots, k+T_p-1\}, \tag{17}$$

$$z_i^b(\kappa + 1) = z_i^b(\kappa) - y_i(\kappa) + y_j(\kappa - \tau_{ji}^b) \quad i, j \in \{1, 2\}, i \neq j,$$
$$\forall \kappa \in \{k, \ldots, k+T_p - 1\}, \tag{18}$$

$$z_i^v(\kappa + 1) = z_i^v(\kappa) + \sum_{j \in \mathcal{R}_i} v_{ji}(\kappa - \tau_{ji}^r) - v_{ij}(\kappa)$$
$$\forall i \in \mathcal{N}, \ \forall \kappa \in \{k, \ldots, k+T_p - 1\}, \tag{19}$$

$$\mathbf{1}_{n_c} u_{ij}(\kappa) \leq v_{ij}(\kappa) \quad \forall j \in \mathcal{R}_i, \ \forall i \in \mathcal{N}, \ \forall \kappa \in \{k, \ldots, k+T_p - 1\}, \tag{20}$$

$$p_i^r \bar{u}_{id}(\kappa) = 0 \quad \forall i \in \mathcal{D}, \ \forall \kappa \in \{k, \ldots, k+T_p - 1\}, \tag{21}$$

$$p_i^d \bar{u}_{di}(\kappa) = 0 \quad \forall i \in \mathcal{D}, \ \forall \kappa \in \{k, \ldots, k+T_p - 1\}, \tag{22}$$

$$\mathbf{1}_{n_c} u_{ij}^b(\kappa) \leq c^b y_i(\kappa) \quad \forall < i, j > \in \mathcal{W}, \ \forall \kappa \in \{k, \ldots, k+T_p - 1\}, \tag{23}$$

$$y_1(\kappa) \in \{0, 1\}, \ y_2(\kappa) \in \{0, 1\} \quad \forall \kappa \in \{k, \ldots, k+T_p - 1\}. \tag{24}$$

## 3 Departure learning

Centralized planning is only possible when one entity has all information and authority to take all decisions. Traditional distributed optimization requires several rounds of communications and introduces artificial fees to shift the local optima. It is often not realistic to assume transport operators will commit to such a scheme. We therefore propose the novel co-planning method departure learning, where the global optimum is the goal of the planning, but only a pre-specified number of potential schedules and the truck operator's expected costs are communicated at each timestep. The method builds on the same assumptions and system dynamics as presented in Sect. 2.1.

At each timestep, the barge operator sends a set $\mathcal{I}(k)$ of barge schedules to the truck operator. The number of schedules is denoted by $n$ and is referred to as the exchanged schedules. The truck operator hereafter computes the transport cost over the prediction horizon for each of the schedules. The costs are send back to the barge operator who adds the costs related only to the barge. The barge operator compares the total costs with the estimated costs of all other feasible schedules and communicates the best schedule to the truck operator. The actions corresponding to the current timestep in the schedule with the best performance are implemented by the barge operator and truck operator separately, and the process is repeated at the next timestep.

To estimate which schedules will perform better, the barge operator uses the performances indicated by the truck operator at previous timesteps to estimate the performance at the current timestep. It is ensured that the set of potential schedules includes both schedules that will perform well and schedules that helps identifying good schedules in the future by using selection strategies that focus on both exploitation and exploration. The overview of the departure learning is shown in Fig. 2. In the following, it is described how the barge operator learns good schedules, and how the truck operator evaluates the cost of a schedule.

### 3.1 Learning good departure times

To estimate what the performance of all schedules are, all schedules must be identified. However, the first departure in a schedule must be from the terminal where the barge currently is, or to which it is travelling. It is thus possible to describe the performance of all feasible schedules if only half the schedules are identified as long as the location of the barge is
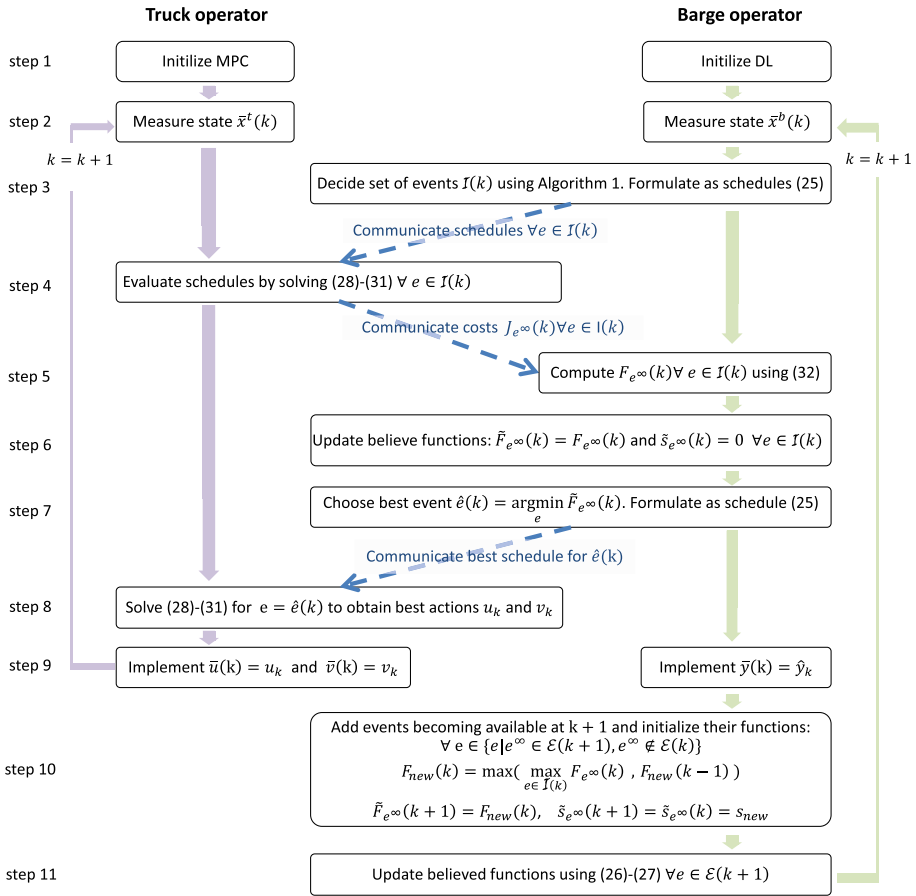
**Fig. 2** Departure learning. Truck operator actions shown as purple flow to the left and barge operator actions as green flow to the right. Blue, dashed arrows indicates the necessary communication

known. This reduces the number of binary options per timestep to one (to depart or not). Such a reduced schedule is called an event $e$. Figure 3 shows an example of a schedule and its corresponding event. Events can be decoded into schedules using the known location of the barge at time $k$ as it determines the first departure terminal and all departures hereafter alternate terminals, mathematically:

$$
y_i(k + \gamma) \le z_i^b(k) + \sum_{\kappa = \tau_{ji}^b - \gamma}^{\tau_{ji}^b} y_j(k - \kappa) \quad \forall \gamma \le \tau_{ji}^b, \forall <i, j> \in \mathcal{W}. \tag{25}
$$

Each element of the event is a binary variable denoted by $b_k$. An event is thus $e = [b_k, \ldots, b_{k+T_p-1}]$ where each element is a specific realizations of $b_k \in \{0, 1\}, \ldots, b_{k+T_p-1} \in \{0, 1\}$. It takes time for the barge to travel between the terminals, and therefore not all events are feasible at all timesteps. The set of events that are feasible at time $k$ is denoted by $\mathcal{E}(k)$. Events at two different timesteps may correspond to the same sequence of events when viewed over an infinite timespan, and

$$\begin{array}{rcl}
[y_1(k) \cdots y_1(k + T_p - 1)] & = & [\,0\;0\;1\;0\;0\;0\,] \\
[y_2(k) \cdots y_2(k + T_p - 1)] & = & [\,1\;0\;0\;0\;1\;0\,] \\
e & = & [\,1\;0\;1\;0\;1\;0\,]
\end{array}$$

**Fig. 3** A schedule consists of two vectors of binary variables describing the departure times from the two end terminals. The corresponding event combines the two

$$\begin{array}{rcl}
e_{a1} \in \mathcal{E}(k) & = & [\,0\;1\;0\;0\;1\;0\,] \\
e_{a2} \in \mathcal{E}(k+1) & = & [\,1\;0\;0\;1\;0\;0\,] \\
e_{a1}^{\infty} = e_{a2}^{\infty} = e_{a}^{\infty} & = & 0\;0\;0\;0\;0\;1\;0\;0\;1\;0\;0\;0\;0\;0\;0\;0 \\
e_b \in \mathcal{N}_{e_a^{\infty}}(k) & = & [\,1\;0\;0\;0\;1\;0\,] \\
e_{c1} \in \mathcal{N}_{e_a^{\infty}}(k) & = & [\,0\;0\;1\;0\;1\;0\,] \\
e_{c2} \in \mathcal{N}_{e_a^{\infty}}(k+1) & = & [\,0\;1\;0\;1\;0\;0\,]
\end{array}$$

**Fig. 4** Illustration of $e_{\infty}$ and $\mathcal{N}_{e^{\infty}}(k)$. Note that the set of neighbours varies over time

are as such identical. $e^{\infty}$ denotes an event over the infinite timespan and is defined as $e^{\infty} = [\mathbf{0}_{1:k} \; e \; \mathbf{0}_{k+T_p:\infty}]$, where $\mathbf{0}_{a:b} = \{0\}^{b-a}$ is a zero-vector of suitable size. If two events are identical except for two subsequent elements, the events are said to be neighbours, i.e. for an event $e_1 = [b_k^1, \ldots, b_{k+T_p-2}^1] \in \mathcal{E}(k)$, the set of neighbouring events is

$$\mathcal{N}_{e_1^{\infty}}(k) = \left\{ e = [b_k^2, \ldots, b_{k+T_p-2}^2] \in \mathcal{E}(k) \,|\, b_i^2 = b_i^1 \; \forall i \backslash \{i = \{j, j+1\} \text{ for exactly one } j \in \right.$$

$\{k, \ldots, k + T_p - 2\}$ for which $b_j^1 = 1$ and $b_j^2 = b_{j+1}^1$, $b_{j+1}^2 = b_j^1\} \backslash \{e_1\}$ This corresponds to two barge schedules only differing in one departure time and for that departure only with one timestep. The set of neighbours are indexed with the event's $e_{\infty}$ and time, since two events $e_1 \in \mathcal{E}(k)$ and $e_2 \in \mathcal{E}(k + 1)$ with $e_1^{\infty} = e_2^{\infty}$ will have the same set of neighbours $\mathcal{N}_{e^{\infty}}$ for all $k$ where $\mathcal{N}_{e^{\infty}} \in \mathcal{E}(k) \cap \mathcal{E}(k + 1)$. Both $e_{\infty}$ and $\mathcal{N}_{e^{\infty}}(k)$ are exemplified in Fig. 4.

The barge operator holds an estimate of the total operation cost for the barge and the truck operator for each event. This estimate of an event's performance is called the event's expected fitness and is denoted by $\tilde{F}_{e^{\infty}}(k)$. If the barge operator has received the operation cost the truck operator will incur if the barge departs according to an event $e$, we say event $e$ has been evaluated. To indicate how certain the expected fitness is, an uncertainty function $\tilde{s}_{e^{\infty}}(k)$ is used. $\tilde{s}_{e^{\infty}}(k)$ decreases when an event corresponding to $e^{\infty}$ or its neighbours are evaluated and increases slowly over $k$. It is expected that the performance indicator for events that share $e^{\infty}$ evolve slowly over time, and that the performance indicators of neighbouring events are related. Like in Bayesian optimization, $\tilde{F}_{e^{\infty}}(k)$ and $\tilde{s}_{e^{\infty}}(k)$ are used to sample a number of candidate events that are expected to either correspond to good barge schedules or provide useful information for the future. Unlike most implementations of Bayesian optimization, the number of feasible events is finite in departure learning, and thus $\tilde{F}_{e^{\infty}}(k)$ and $\tilde{s}_{e^{\infty}}(k)$ can be computed for all events.

The set of candidate events $\mathcal{I}(k)$ is sampled using strategies based on ranking of $\tilde{F}_{e^{\infty}}(k)$, $\tilde{s}_{e^{\infty}}(k)$ and functions of the two, together with random selection as outlined in Algorithm 1 for balanced exploitation and exploration. The cardinality of $\mathcal{I}(k)$, denoted by $n$, is the number of schedules the truck operator must evaluate. Notice that the cost of each schedule is independent of the other schedules and the operator therefore can evaluate the schedules in parallel.

After the barge operator receives the cost of each evaluated event from the truck operator, the expected fitness of these events are updated and their uncertainty values are set to zero.

---

**Algorithm 1** The strategy used to assemble $\mathcal{I}(k)$

---

1: **input** $\tilde{F}_{e\infty}(k)$, $\tilde{s}_{e\infty}(k)$, $\mathcal{E}(k)$
2: **return** $\mathcal{I}(k)$ with $n$ unique events
3: $\mathcal{I}(k) = \emptyset$
4: **for** $i \leftarrow 1$ to $floor(n/6)$ **do**
5:　$\mathcal{I}(k) = \mathcal{I}(k) \cup \arg\min_{e\in\mathcal{E}(k)\setminus\mathcal{I}(k)} \tilde{F}_{e\infty}(k) + \tilde{s}_{e\infty}(k)$
6:　$\mathcal{I}(k) = \mathcal{I}(k) \cup \arg\min_{e\in\mathcal{E}(k)\setminus\mathcal{I}(k)} \tilde{F}_{e\infty}(k)$
7:　$\mathcal{I}(k) = \mathcal{I}(k) \cup \arg\max_{e\in\mathcal{E}(k)\setminus\mathcal{I}(k)} \tilde{s}_{e\infty}(k)$
8:　$\mathcal{I}(k) = \mathcal{I}(k) \cup \arg\min_{e\in\mathcal{E}(k)\setminus\mathcal{I}(k)} \tilde{F}_{e\infty}(k) - \tilde{s}_{e\infty}(k)$
9:　**for** $j \leftarrow 1$ to $2$ **do**
10:　　$\mathcal{I}(k) = \mathcal{I}(k) \cup \text{rand}\,(e \in \mathcal{E}(k) \setminus \mathcal{I}(k))$
11:　**end for**
12: **end for**
13: **for** $i \leftarrow floor(n/6)6$ to $n$ **do**
14:　$\mathcal{I}(k) = \mathcal{I}(k) \cup \text{rand}\,(e \in \mathcal{E}(k) \setminus \mathcal{I}(k))$
15: **end for**

---

Some events will be feasible at the next time $k + 1$ which were not feasible at time $k$. These events are initialized with the maximum fitness evaluated at $k$ and the uncertainty value $s_{new}$. Hereafter, all the fitness and uncertainty values of all events are updated as follows:

$$\tilde{F}_{e\infty}(k+1) = \alpha \tilde{F}_{e\infty}(k) + \frac{1-\alpha}{|\mathcal{N}_{e\infty}(k) \cup \mathcal{N}_{e\infty}(k+1)|} \sum_{i\in\mathcal{N}_{e\infty}(k)\cup\mathcal{N}_{e\infty}(k+1)} \tilde{F}_i(k) \qquad (26)$$

$$\tilde{s}_{e\infty}(k+1) = (\alpha+\beta)\tilde{s}_{e\infty}(k) + \frac{1-\alpha}{|\mathcal{N}_{e\infty}(k) \cup \mathcal{N}_{e\infty}(k+1)|} \sum_{i\in\mathcal{N}_{e\infty}(k)\cup\mathcal{N}_{e\infty}(k+1)} \tilde{s}_i(k) \quad (27)$$

The learning parameter $\alpha$ balances the emphasis laid on each events' previous value and on neighbouring events' values and the factor $\beta$ controls the speed at which information from previous timesteps become uncertain. To initialize departure learning prior knowledge can be used, otherwise it is recommended that $\tilde{F}_{e\infty}(1) = \tilde{F}_{init} \; \forall\, e \in \mathcal{E}(1)$ where $\tilde{F}_{init}$ is higher than the expected maximum fitness and $\tilde{s}_{e\infty}(1) = s_{new} \; \forall\, e \in \mathcal{E}(1)$. $s_{new}$ is the maximum uncertainty and is also used to update new feasible events in step 10 of the method overview in Fig. 2.

### 3.2 Evaluating the performance

The truck operator evaluates the performance of the communicated schedules by planning container and truck routes simultaneously for each $e \in \mathcal{I}(k)$. To do so he solves the following optimization problem, initiated from the current state for the given schedule:

$$J_{e\infty}(k) = \min \sum_{\kappa=k}^{k+T_p-1} J^t(\kappa) \qquad (28)$$

$$\text{s.t. } x^t(k) = \bar{x}^t(k) \qquad (29)$$

$$\{< y_1(\kappa), y_2(\kappa) > | \kappa \in \{k, \ldots, k+T_p-1\}\} = e \qquad (30)$$

$$(14), (16), (17), \text{ and } (19)-(23). \qquad (31)$$

After receiving the truck operator's cost for an event, the barge operator adds its private costs to compute the total predicted cost which serves as the event's fitness, i.e.

$$F_{e\infty}(k) = J_{e\infty}(k) + \sum_{\kappa=k}^{k+T_p-1} J^b(\kappa) : \{< y_1(\kappa), y_2(\kappa) > |\kappa \in \{k, \ldots, k+T_p-1\}\} = e. \tag{32}$$

## 4 Simulation experiments

When departure learning is used to co-plan barge departures, the learning rate of the method and the realised cost are dependent on departure learning's four tunable parameters: the prediction horizon, $T_p$; the learning parameter, $\alpha$; the forgetfulness parameter, $\beta$; and the number of communicated schedules, $n$. These dependencies were investigated numerically in simulated experiments. A well-tuned departure learning was hereafter compared to the performance of a method without cooperation, the centralized method presented in Sect. 2.1, and a co-planning method without learning. In all experiments it is assumed that decisions are taken every $\Delta t = 15$ min. In this section, the used benchmark methods and the scenarios are first described in detail. Second, the impacts of the tunable parameters are presented. Finally, the departure learning and the three benchmark methods are compared. All experiments are performed in Matlab formulated with Yalmip (Löfberg, 2004) and solved by Gurobi.

### 4.1 Benchmark methods

The performance of departure learning is benchmarked against three methods: (1) the *centralized method*, presented in Sect. 2.3, which requires full cooperation and unlimited information sharing, (2) a *fixed method* that does not require any cooperation and (3) an *uninformed* co-planning *method* without memory of previous plans.

The fixed method that requires no cooperation mimics the traditional division between decisions taken at the tactical and the operational level, while still assuming a-modal bookings. In this method, a pre-defined barge schedule is used and thus only trucks and containers can be routed in real-time. The used schedule has as many barge departures as possible, since the likelihood that a container can be transported by barge and arrive in time is higher if there are more departures. If a schedule with fewer departures was used, the fixed schedule would have a known disadvantage in terms of late delivery, which we want to avoid to evaluate the added value of departure learning. During the simulation, the truck operator solves the truck and container planning problem (28)–(31) every $\Delta t$ min in an MPC fashion using this publicly available schedule.

The uninformed co-planning method follows the steps outlined in Fig. 5. The method deviates from departure learning in steps 3, 6, 7, 10 and 11 from Fig. 2. Instead of using Algorithm 1 to actively choose which schedules to propose to the truck operator, the barge operator sends $n$ randomly chosen schedules from the set of feasible schedules. Step 10 and 11 are omitted and step 6 and 7 are replaced by one step where the best of the schedules evaluated at the current timestep $k$ is decided to be implemented.

### 4.2 Scenarios

The numerical experiments were performed on the Dutch network shown in Fig. 1 where Rotterdam and Apeldoorn are origins and destinations and Nijmegen is a terminal for trans-
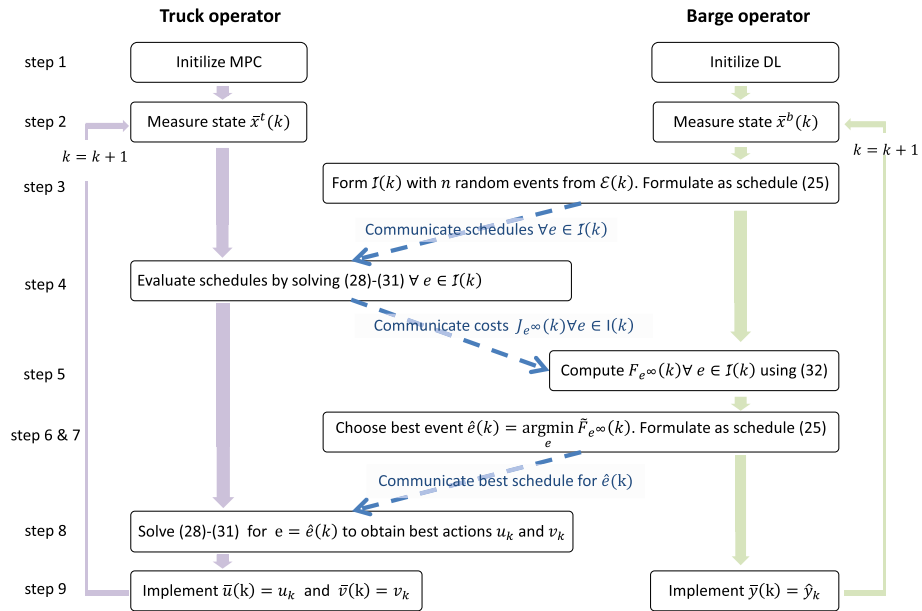
**Fig. 5** Actions of the uninformed method

shipments. The network thus accommodates $n_c = 2$ different commodities: *import* to be transported from Rotterdam to Apeldoorn, and *export* to be transported in reverse direction. Departure learning and the benchmark methods all scale to larger truck networks. The three-node network was chosen to keep the computation time short, the results easily interpretable, and in order to be able to clearly focus on the co-planning components. It is assumed that trucks drive 90 km/h and (un)loading a truck in Rotterdam takes 20 min, while it is 10 min in Nijmegen and Apeldoorn. With these assumptions, the 140 km distance between Rotterdam and Apeldoorn corresponds to 123 min travel time, and the 55 km distance between Nijmegen and Apeldoorn takes 56 min. The barge between Dordrecht and Nijmegen is by Qu et al. (2019) reported to take 5 h including loading, so we assume the total travel time between Rotterdam and Nijmegen is 6 h. The capacity of the barge is assumed to be 100 containers. The truck operator is assumed to have 36 trucks, each of which can transport one container. They are all parked in Apeldoorn at the beginning of the simulation. The barge departure schedule used by the fixed method lets the barge sail between Rotterdam and Nijmegen as frequently as possible. The first barge departs at timestep 1 from Nijmegen and departs hereafter alternating between the two terminals every 6.5 hr corresponding to $\tau_{12}^b + 2 = 26$ timesteps.

The transport cost is in most of the literature on synchromodal transport computed primarily from the shippers perspective (Wenhua et al., 2019; Resat & Turkay, 2019; Behdani et al., 2016). This does not capture the cost of repositioning empty trucks and under-full barges realistically. One strength of departure learning is the ability to track empty vehicles and thus the ability to assign cost to them, see Larsen et al. (2020). We therefore use the vehicle-centred costs shown in Table 1.
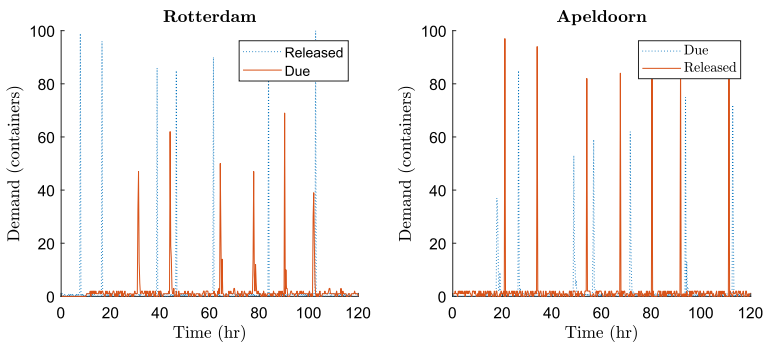
The demand used in the experiments contains both import and export. Several different demand profiles were used. A summary can be seen in Table 2 and Fig. 6 shows the demand profile with high peaks. In the hinterland of container ports, the large quantities of containers

**Table 1** Parameters and costs of the realistic scenario

| Non-zero initial states | Network parameters | Costs | |
|---|---|---|---|
| $\bar{z}_2^b(0) = 1$ | $\tau_{12}^b = \tau_{21}^b = 24$ | $w_{12}^b = w_{21}^b = 210$ | $w_d = 1000$ |
| $\bar{z}_2^v(0) = 36$ | $\tau_{13}^r = \tau_{31}^r = 9$ | $w_{13}^v = w_{31}^v = 73.19$ | |
| | $\tau_{23}^r = \tau_{32}^r = 4$ | $w_{23}^v = w_{32}^v = 33.93$ | |
| | $c^b = 100$ | $w_{12}^l = w_{21}^l = 13.18\ \mathbf{1}_{n_c}$ | |

**Table 2** Summary of the used demand profiles

| | Release Rotterdam | Release Apeldoorn | Peak height (of barge capacity) |
|---|---|---|---|
| High peaks | 2027 | 2324 | 80–100% |
| Medium-high peaks | 1811 | 1868 | 60–80% |
| Medium-low peaks | 1338 | 1824 | 40–60% |
| Low peaks | 869 | 1406 | 20–40% |
| Unbalanced base quantity | 501 | 1461 | – |
| Unbalanced medium-high quantity | 1441 | 2425 | – |



**Fig. 6** The first 5 days of the demand profile with high peaks used in the experiments

that are to be loaded and unloaded from one ship in a relatively short amount of time make the demand profiles with peaks very realistic and relevant to practitioners. In scenarios without significant peaks in the demand, it is likely that there is a predominant direction of transport of full containers, making the unbalanced demand profiles a realistic type of demand to contrast the demand with peaks. Both profiles are used in order to showcase the strengths and weaknesses of departure learning. For all demands with peaks, a base-demand of 0 to 2 containers are released in Apeldoorn and 0 to 1 containers in Rotterdam every 15 min. On top of this, a random number of containers within the indicated interval of the barge capacity are released at both locations at independent and irregular time intervals between 7 and 22.5 hr. In the unbalanced demand with base quantity, between 0 and 1 containers are released at Rotterdam and between 0 and 3 are released at Apeldoorn every 15 min. For unbalanced demand with medium-high quantity, between 0 and 3 containers are released at Rotterdam and between 1 and 4 are released at Apeldoorn every 15 min. The profiles were constructed such that each container was released at least 10 hr before they were due at the other location. The number of containers due at a virtual demand node is drawn at each

timestep from a uniform distribution between zero and the number of containers that can have due date at this destination at this time. Notice the profiles have different total demand quantity. The demand was sampled once for each profile and used in all experiments with that profile.

The impact of the tunable parameters are shown with simulations using the demand with high peaks and the unbalanced demand with a base quantity of containers as these two profiles are the extremes in terms of demand volatility and quantity. Results from all demand profiles are used in the final comparison between departure learning and the benchmark methods.

When appropriate, the experiments are a simulation of 5 days transport in the system where, initially, no containers are present and all trucks are parked in Apeldoorn. This simulation setup will be referred to as the *long simulation setup*. A *short simulation setup* has also been used. Experiments with this setup and the demand profiles with high peaks start with the state of the system after the centralized method with $T_p = 80$ has been used for 31,75 hr (127 timesteps) on the long simulation setup, while those on the unbalanced demand with base quantity use that of the centralized method after 34,25 hr (137 timesteps). These experiments stop 101 timesteps after. This time period is chosen since it starts after the demand profile with peaks is fully established and covers a time period where the realized cost when using the optimal method is higher than average, which indicates that the problem is more complex during this period.

### 4.3 Impact of the tunable parameters

The impact of the tunable parameters has been investigated on a series of experiments, where conclusions made on earlier experiments impacted tuning decisions on later ones. The experiments were primarily performed with the demand profile with high peaks as the demand peaks make the difference between *good* and *bad* departure times very visible and thus provide clearer explanation. Some experiments are repeated with the unbalanced demand profile with base quantity. In the following sections, the impact of each tunable parameter will thus be presented after an introduction and a description of the experiments that lead to the insights. In all sections, time will be indicated as timesteps and to initialize new events $s_{new} = J_{new} = 10^7$ was used.

### 4.3.1 Prediction horizon—$T_p$

The prediction horizon impacts not only departure learning, but also the benchmark methods since they are all MPC-based. The longer the prediction horizon is, the more information each method will have available to optimize the cost. The optimization problem only sees the advantage of moving a vehicle or container, if the prediction horizon is longer than the travel time. We therefore have considered only prediction horizons longer than $T_p = \tau_{12}^b + \tau_{23}^r = 28$ timesteps. To ensure the random variables in departure learning and the uninformed method or the choice of schedule for the fixed method do not impact the results, the centralized method was used to show the impact of the prediction horizon. The long simulation setup with the high peaks demand profile was used in this experiment.

The results in Fig. 7 show that as the prediction horizon increases, the total cost of the realised actions decreases. It is noteworthy that the realised cost is significantly higher when the prediction horizon is too short to foresee the implications on the containers of a round-trip of the barge. The results, furthermore, show that longer prediction horizons increase the computation time significantly. Since the method is MPC-based, the optimization problem
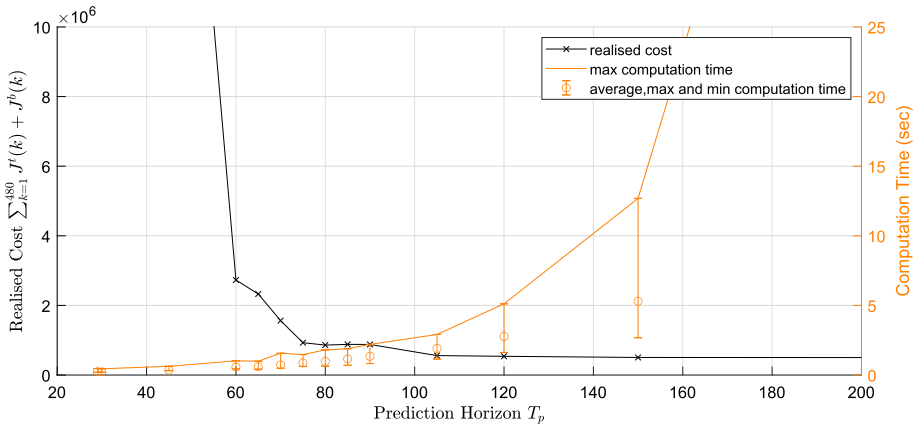
**Fig. 7** The trade-off between the total cost realised in the simulation and computation time. With increasing prediction horizon, the realised cost decreases and the time it takes to solve the optimization problem each timestep increases. Long simulation setup, demand with high peaks, centralized method

is solved at each timestep. In the figure, the shortest, longest and average computation times for solving the optimization problem one time are reported. The maximum computation time determines the speed of the method; $\Delta T$ must be higher than the slowest computation to ensure new decisions are available at all timesteps. This value rises quickly with increasing prediction horizon. For the remainder of the experiments in this paper the prediction horizon is $T_p = 80$, as it is a reasonable trade-off between the achievable realised cost and the computation time.

### 4.4 Exchanged schedules—$n$

The more schedules the barge operator gets feedback on from the truck operator, the more information is available to decide on departures and future communication. However, for each schedule communicated, the truck operator will have to optimize the planning problem (28)–(31). This can be done in parallel to decrease the computation time if the truck operator has sufficient parallel computation capacity. With each schedule communicated, the barge operator gets a little more insight into the truck operator's cost structure and current demand profile since no natural noise from the shifting demand profiles is present. Therefore, it is desirable for the truck operator, both from a computation and an information perspective, to provide feedback on the lowest number of schedules that can ensure satisfactory barge departures.

The statistical information on the realised cost for five repetitions of using departure learning and the uninformed method with different numbers of exchanged schedules on the short simulation setup are in Fig. 8 compared to the realized cost obtained by the centralized method. Especially the uninformed method benefit from exchanging more schedules, but the performance of departure learning does also improve. This is expected since the probability of randomly choosing schedules that results in a good realized performance increases when more schedules are exchanged. The results shows that the uninformed method is more sensitive to this effect than departure learning. For all considered $n$ it is clear that departure learning deliver better and more consistent results than the uninformed method. The remaining experiments will be performed with $n = 6$ exchanged schedules.
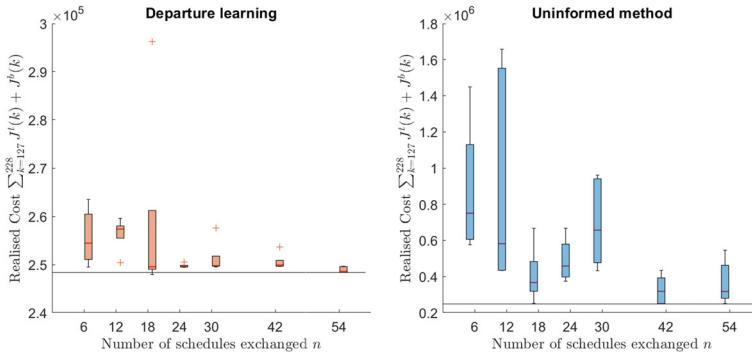
**Fig. 8** The box-plots show the min, max, 25th, 50th and 75th percentile of the realized cost when departure learning ($\alpha = 0.7$ and $\beta = 0.1$) and the uninformed method is exchanging different numbers of schedules. Red + indicates outliers. The black line is the realized cost when using the centralized method. Notice the scale difference. Short simulation setup, 5 repetitions, demand with high peaks, $T_p = 80$
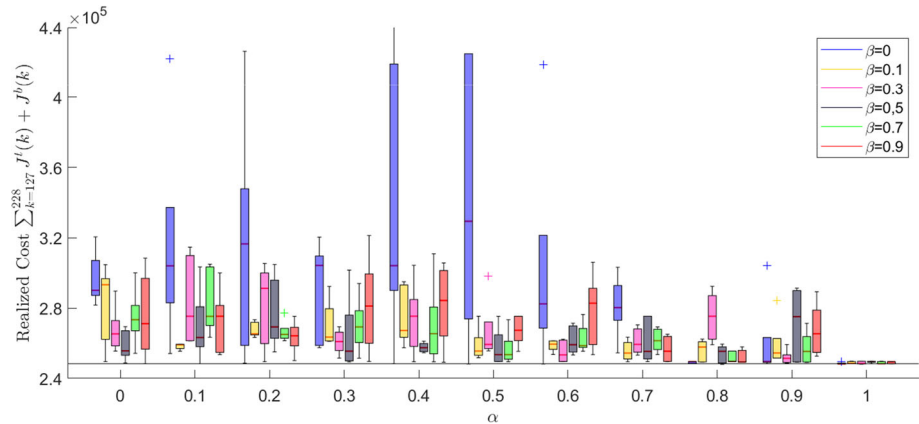


**Fig. 9** The statistics of the realized cost using departure learning with different combinations of $\alpha$ and $\beta$. Each group of box-plots shows the statistics for departure learning using $\alpha$ as indicated on the horizontal axis. Within a group, the color indicates the used $\beta$-value as indicated by the legend. The leftmost box-plot in a group is $\beta = 0$. The black line is the realized cost achieved by the centralized method. Short simulation setup, 5 repetitions, demand with high peaks, $T_p = 5, n = 6$

## 4.5 Learning parameters—$\alpha$ and $\beta$

Departure learning's ability to learn good schedules is tightly linked to the update of the believed fitness and the uncertainty, (26) and (27). These updates are highly dependent on the learning parameters $\alpha$ and $\beta$. These parameters have no impact on the communication between the barge and truck operator, neither on the computation time. To investigate the impact, experiments using departure learning with different combinations of $\alpha$ and $\beta$-values were performed on the small simulation setup with $T_p = 80$ and $n = 6$. Each experiment was repeated five times.

The realized cost of each experiment is shown in Fig. 9. Departure learning with all combinations of $\alpha$ and $\beta$ perform better than the uninformed method, which, as seen in Fig. 8, makes the total transport cost minimum €576,383. In four instances, the smallest
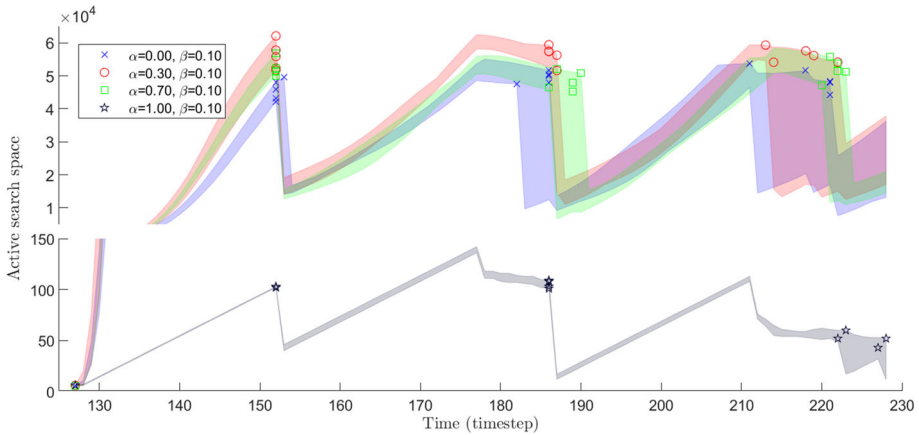
**Fig. 10** Departure learning's active search space over time for four different tunings of $\alpha$ and $\beta$. The borders of the coloured panels are min and max over the repetitions of the experiment. The realized departure times and corresponding search space is for each repetition indicated by a marker. Notice the gap in the vertical axis. Short simulation setup, 5 repetitions, demand with high peaks, $T_p = 5, n = 6$

realized cost obtained using departure learning was smaller than the cost of the centralized method, and in another four instances using $\alpha = 1$ even the 25th percentile was smaller. This happens because the departure learning's at some timesteps implement actions that at that timestep seem suboptimal, but over time open up for decisions which improves the realized cost.

It is clear from the results that departure learning with $\alpha = 1$ performs differently regardless of $\beta$. When $\alpha = 1$ the expected fitness of each event is only updated based on that event's earlier evaluations. Figure 10 shows the number of events that has an expected fitness different from the initialization of the event. In other words, it shows how many events departure learning has an expectation about, and thus implicit how wide the majority of the search is. We call this number the *active search space*. In the figure, the realized departure times for each repetition are also marked. All repetitions with all tunings of departure learning departs the barge at the simulation's first timestep $k = 127$ because of the implementation method using Matlab sort function and because this departure also for the centralized method is optimal. When departure learning implements a barge departure, the active search space collapses rapidly to a significantly smaller size. When the barge departs, all schedules with departure times within the travel time become infeasible. The large collapses just after a departure is realized thus indicate that departure learning had investigated several events with departure at the realized time or soon after. Slow decreases in the active search space occur when schedules with a departure at that timestep becomes obsolete since the barge did not depart.

Departure learning with $\alpha = 1$ has, in addition, rapid collapses in the active search space $\tau_{12}^b = 24$ timesteps after each departure, regardless of whether there is a new departure or not. For departure learning with $\alpha \neq 1$ this effect is also visible, but the decrease is less rapid. This indicates, that departure learning with $\alpha = 1$ to a higher extend focus on the same plan without investigating plans that has slightly different departure times, and thus becomes obsolete at different timesteps. This is furthermore supported by the very small active search space. It is thus likely that even though $\alpha = 1$ is a very good tuning for the investigated scenario, it may perform poorly in other cases or if different plans are found initially.
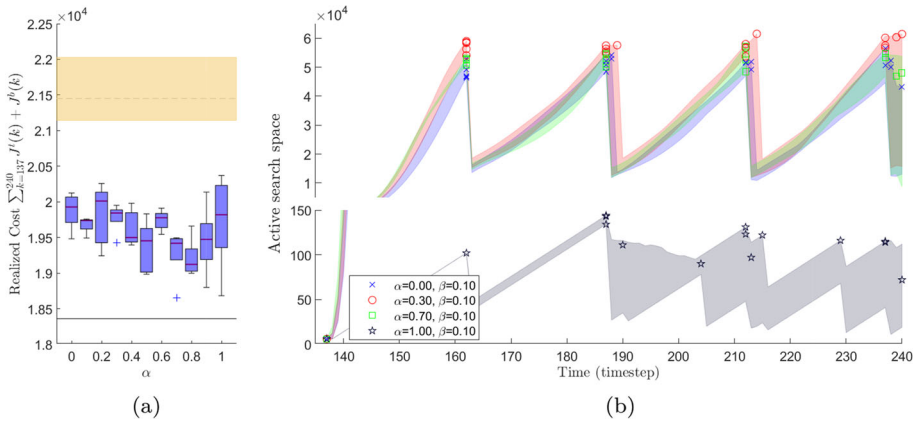
**Fig. 11** The realized cost and search space for departure learning for experiments on the unbalanced demand profile. **a** corresponds to Fig. 9 and **b** to Fig. 10. Notice the total demand is different. In all experiments $\beta = 0.1$. The yellow area in **a** is the min, max and average realized cost obtained by the uninformed method. Small simulation setup, 5 repetitions, unbalanced demand with base quantity, $T_p = 5$, $n = 6$

In scenarios where the exact departure time is of less importance, and the centralized method suggests that very frequent barge departures yield the best performance, the same tendencies for $\alpha = 1$ are seen. Figure 11a shows that departure learning with this profile still performs better than the uninformed profile, but not as pronounced as with the demand profile with peaks. It furthermore shows that departure learning with $\alpha = 1$ has a very large variance. This can again be explained by the very narrow search space in combination with the relative performance of the uninformed method.

When there are peaks in the demand, the difference between choosing random and learned schedules is higher. This indicates that the time of departure is more important. For the unbalanced demand profile, the centralized method departs as frequently as possible. The first schedule departure learning learns thus remain a good schedule in the beginning of the simulation. However, this schedule only has three departures because of the prediction horizon length. When the simulation approaches the third departure, it is thus likely that one of the randomly chosen schedules in $\mathcal{I}(k)$ will perform better, leading to a large diversity in the realized departure times. The centralized method only departs three times when used on the demand profile with peaks. These departures correspond to the departure found by departure learning with $\alpha = 1$. For this profile, it is not as important if the schedule that are planned around the third departure include other departures in the future or not, and it is thus likely that departure learning with $\alpha = 1$ can find this last "optimal" departure. When the unbalanced demand is used, the difference between the departure times has less impact and it is thus likely that departure learning with $\alpha = 1$ finds different good departures at each repetition of the experiment. In conclusion, it is not recommended to use departure learning with $\alpha = 1$ since the first chosen schedules will have a very large impact on the future realized cost.

Returning to Fig. 9, a clear pattern in the impact of $\alpha$ and $\beta$ is not visible. There is a tendency that higher $\alpha$-values in combination with lower $\beta$-values gives better results. For $\alpha \leq 0.7$ and lower, departure learning with $\beta = 0$ performs very poorly. This is likely because when $\beta = 0$, equal confidence is put on evaluations performed in the recent and distant past. When new information becomes known by departure learning, earlier evaluations

may become obsolete. Very high $\alpha$ likely compensates for this effect with the increased focus on previously evaluated schedules. For the final comparison between departure learning and the benchmark methods, $\alpha = 0.7$ and $\beta = 0.1$ are chosen.

## 4.6 Comparison between departure learning and the benchmark methods

The right tuning of departure learning can improve the method's performance, as seen in the previous sections. In this section departure learning with $\alpha = 0.7$ and $\beta = 0.1$ is compared to the three benchmark methods. Key results from the simulations are presented in Table 3. Values in rows marked with (1000) should be multiplied by 1000 for the correct representation.

The comparison is done on the results from the long simulation setup with $T_p = 80$ for all methods and for all demand profiles. Two versions of the uninformed method was used for the experiments with demand with high peaks and with unbalanced demand with base quantity, one with $n = 6$ and one with $n = 42$. Departure learning uses $n = 6$. The experiments with departure learning and the uninformed method are repeated five to ten times and average values over the repetitions are reported.

The centralized method performs, as expected, best in terms of realized cost as one decision maker has access to all information and thus can coordinate the barge and trucks optimally. Realistic methods with limited communication between the truck and barge operators can only perform as well as a centralized method in carefully curated scenarios. Under the scenarios where the demand has peaks, departure learning performs very well compared to the fixed method. In scenarios with unbalanced demand, the fixed method outperforms departure learning with a small margin.

Under both profiles with unbalanced demand, the centralized method uses many barge departures (17 and 19). The used fixed schedule has 19 departures, which means the accessibility of barge transport capacity is either higher or the same under the fixed schedule as under the best possible method. Departure learning does not have good information of all possible schedules as only few schedules are exchanged ($n = 6$). As a barge departure only happens if a schedule with an immediate barge departure is expected to be the most beneficial, departure learning will depart fewer barges in scenarios where there is a high chance that a schedule with its first departure is in the future will imply a similar expected cost. This causes departure learning to depart the barge less than the centralized method in the scenarios where the demand is unbalanced and thus do not have peaks to clearly define beneficial departure times. This indicates that departure learning will provide benefits in many port hinterlands and scenarios where the timing of the consolidation of freight is important. In scenarios with a very predictable, steady demand, a fixed schedule that is optimized to the upcoming demand will perform better. In all scenarios the uninformed method shows the poorest performance. Even with $n = 42$ exchanged schedules per timestep, the performance is worse than departure learning with only $n = 6$ exchanged schedules. The relative difference in the performance of the different methods grows with larger peak height.

In Fig. 12 the departures realized by each method for the demand with high peaks are seen. In the later half of the simulation, the fixed method's schedule is very misaligned with the "optimal" schedule provided by the centralized method, while departure learning remains similar. Departure learning departs at relatively similar times in the different repetitions. Corresponding plot for the other demand profiles clearly shows that the higher the correlation between departure time and the realized costs is, the more likely departure learning is to outperform both the fixed and the uninformed method.

**Table 3** Performance of departure learning and the three benchmark methods on the long simulation setup, $T_p = 80$. Departure learning uses $\alpha = 0.7$, $\beta = 0.1$, $n = 6$

|  | Departure learning | Centralized | Fixed | Uninformed $n = 6$ | Uninformed $n = 42$ |
|---|---|---|---|---|---|
| *Demand with high peaks* | | | | | |
| Realized costs (1.000) | 2068 | 860 | 7936 | 8587 | 3815 |
| Unsatisfied demand | 1922 | 720 | 7787 | 8439 | 3668 |
| Barge departures | 16.6 | 17 | 19 | 12.8 | 13.8 |
| Barge utilization | 35.1% | 42.8% | 21.9% | 28.1% | 35.2% |
| Truck departures | 2285 | 2281 | 2195 | 2174 | 2217 |
| Truck utilization | 83.6% | 84.0% | 85.8% | 87.0% | 85.8% |
| *Demand with medium-high peaks* | | | | | |
| Realized costs (1.000) | 162.0 | 116.8 | 261.6 | 525.8 | |
| Unsatisfied demand | 38 | 0 | 134 | 398 | |
| Barge departures | 14.5 | 14 | 19 | 11.5 | |
| Barge utilization | 23.1% | 43.2% | 14.2% | 20.0% | |
| Truck departures | 1886 | 1877 | 1865 | 1902 | |
| Truck utilization | 87.2% | 87.0% | 87.9% | 87.7% | |
| *Demand with medium-low peaks* | | | | | |
| Realized costs (1.000) | 119.9 | 112.3 | 170.9 | 229.2 | |
| Unsatisfied demand | 1 | 0 | 51 | 107 | |
| Barge departures | 17.4 | 15 | 19 | 11.2 | |
| Barge utilization | 17.1% | 31.5% | 13.0% | 15.0% | |
| Truck departures | 1803 | 1792 | 1787 | 1809 | |
| Truck utilization | 83.7% | 83.3% | 83.8% | 84.0% | |
| *Demand with low peaks* | | | | | |
| Realized costs | 84,377 | 81,957 | 87,523 | 129,116 | |
| Unsatisfied demand | 0 | 0 | 0 | 38 | |
| Barge departures | 17.5 | 15 | 19 | 9.1 | |
| Barge utilization | 20.5% | 32.0% | 14.8% | 20.4% | |
| Truck departures | 1360 | 1379 | 1349 | 1420 | |
| Truck utilization | 80.9% | 81.4% | 81.5% | 81.4% | |
| *Unbalanced demand with base quantity* | | | | | |
| Realized costs | 85,132 | 81,825 | 84,066 | 91,919 | 89,616 |
| Unsatisfied demand | 0 | 0 | 0 | 0 | 0 |
| Barge departures | 14.8 | 17 | 19 | 9.4 | 9.2 |
| Barge utilization | 23.2% | 25.1% | 22.0% | 24.6% | 26.0% |
| Truck departures | 1427 | 1431 | 1445 | 1460 | 1434 |
| Truck utilization | 66.9% | 66.8% | 66.7% | 67.1% | 66.9% |

**Table 3** continued

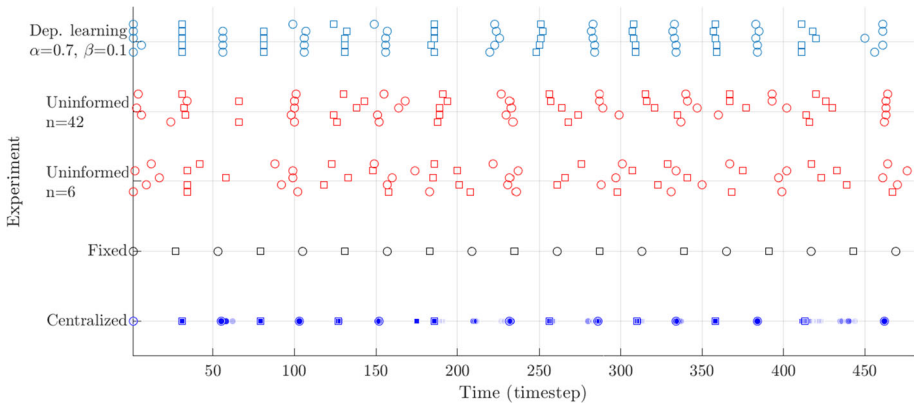|  | Departure learning | Centralized | Fixed | Uninformed $n = 6$ | Uninformed $n = 42$ |
|---|---|---|---|---|---|
| *Unbalanced demand with medium-high quantity* | | | | | |
| Realized costs (1.000) | 488.0 | 143.0 | 447.4 | 6 550.4 | |
| Unsatisfied demand | 331 | 0 | 282 | 6327 | |
| Barge departures | 17.5 | 19 | 19 | 12.0 | |
| Barge utilization | 30.2% | 35.5% | 27.1% | 25.7% | |
| Truck departures | 2352 | 2390 | 2345 | 2290 | |
| Truck utilization | 79.0% | 78.4% | 78.8% | 79.9% | |



**Fig. 12** The departures realized by each of the four methods. Circular markers represent departures from Apeldoorn and squares from Rotterdam. For departure learning and the uninformed method, the results of the five repetitions are reported above each other. The departures, the central method considered but did not implement, are shown as smaller, transparent markers. The intensity of the color of these markers thus indicates how long the central method considered a departure beneficial. Long simulation setup, 5 repetitions, demand with high peaks, $T_p = 5$

Departure learning and the benchmark methods all try to reduce the total realized cost. When doing so, they achieve different usage of the barge and trucks. When a fixed schedule is being followed, the barge departs 19 times during the experiment. This number varies for the other methods. When the demand has high peaks, departure learning uses a similar number of barge departures as the centralized method, indicating that a large part of the good performance departure learning reaches in these cases is from correctly chosen departures. The barge departs less when the uninformed method is used, which is expected as the chance that a randomly chosen schedule with a departure in the first-coming timestep is exchanged at a suitable time is relatively low.

The utilization of the barge capacity is, as expected, best for the centralized method in most scenarios. The uninformed methods gain on average better results when $n = 42$ on unbalanced demand with base quantity. However, the utilization of the barge varies for the uninformed method without a clear pattern. This is likely due to the utilization of the barge being tied to the departure moment of the barge, which as discussed above is less predictable for the uninformed method. When $n = 42$, we see a higher barge utilization than when $n = 6$

which supports that the more schedules are exchanged, the higher is the chance that a suitable one with immediate departure is among them. Departure learning does in all cases result in a higher barge utilization than the fixed schedule. The number of truck departures and the truck utilization does not vary significantly between the different methods.

In all scenarios but one, the centralized method manages to deliver all containers with due time within the simulation period in time. The reported unsatisfied demand is measured in containers times timesteps. The tendencies between the methods are very similar to those observed for the realized costs. For demand with peaks, departure learning performs much better than the fixed method which again outperforms the uninformed methods. When the demand is low, all methods manages to deliver the containers in time, and as the quantity increases, the late deliveries also increase. It is interesting to notice that while the unbalanced demand profile with medium-high quantity has 5% higher demand quantity than the profile with medium-high peaks, the unsatisfied demand is 110% higher for the fixed method and 771% higher for the unbalanced demand. At the same time, the number of trucks and barge departures to transport the demand increases for all methods and the truck utilization decreases. For the centralized method, the barge utilization is lower for the unbalanced demand, while it for departure learning and the fixed method increases. These factors indicate that the main driver of delays and costs in the scenario with unbalanced medium-high demand quantity is the scarcity of trucks to compensate for sub-optimal barge schedules.

In conclusion, for the demand profile with peaks, departure learning is a very good method for systems where the barge and truck planning cannot be integrated due to stakeholder interests. blue For situations with unbalanced demand, where high frequency of barge departures is more important than integration of plans, departure learning performs slightly worse than the fixed method if it is constructed with frequent departures. Systems with different characteristic will benefit from different approaches, but departure learning is a promising method if the system has higher correlation between barge departure times and costs.

## 5 Conclusions and future research

The efficiency in the logistics sector can improve significantly if the involved stakeholders cooperate in real-time. However, cooperation between competitors requires co-planning methods that not only give the cooperating partners an advantage towards external competition but also protect the partners from losing information, clients and autonomy to one another. In this paper, we present a novel method, called *departure learning*, which facilitates real-time co-planning of barge schedules between a barge and a truck operator. We show that departure learning is a promising method for cooperation in practice and how the tunable parameters of departure learning affects the performance of the combined transport system. When the transport system is under pressure and consolidation of demand is only possible at specific barge departure times, departure learning outperforms the current practice where schedules are fixed ahead of time. The more information there is available to each operator when planning, the cheaper the realization will be. The computation time does, however, increase significantly. Less feedback from the truck operator on the barge operator's departure plans decreases the possibilities for inferring information. It was found that even with feedback on only 6 schedules at each timestep, departure learning's performance was sufficient to achieve good performance. The expected performance of all schedules is updated using two learning parameters. The results show that extreme parameters limit departure learning and that higher $\alpha$ values in combination with lower $\beta$ values perform well. It was

shown that, regardless of the parameters used, departure learning was superior to receiving feedback on random schedules without remembering previous information.

There are several important directions for further research. Departure learning is built on ideas from Bayesian optimization and we have shown the impact of tuning one application of these ideas. Research into variations of the decisions taken when applying the ideas could reveal other promising methods. It would especially be interesting to consider what impact the balance between exploration and exploitation has; if other definitions of neighbouring events can provide benefits; and how prior information on expected good departure times can be incorporated. The latter will improve the performance of departure learning when disruptions occur, as it can mitigate the assumption that information obtained earlier is still applicable to a high degree. Other interesting research directions are investigating other learning strategies' strengths and weaknesses for co-planning.

The strength of departure learning is the clear division of authority and responsibility and the very minimal information exchange. This makes the method more applicable in practice than many current, academic, cooperation methods. The presented formulation of departure learning does, however, build on a number of limiting assumptions. Among these, the limited number of stakeholders involved, the two-terminal barge route, and the lack of operation time considerations stand out as immediate targets for further research. It is certain that the method can work when multiple truck operators cooperate with one barge, but how this impacts the learning parameters and how fairness is ensured are still open questions. It would furthermore be interesting to apply departure learning to a case study and focus on a profit sharing scheme that both encourages participation in departure learning and discourages malicious behaviour.

Departure learning shows that it is possible to co-plan with very limited information sharing and no loss of autonomy. In contrast to our current, hierarchical, transport planning system, departure learning can adapt departure times to real-time information. With further research it can become a practical tool for transport operators to increase the utilization of their transport capacity and thus help alleviating the negative impacts on the environment.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

# Appendix A

**Table 4** Notation

| | |
|---|---|
| $t$ | Time |
| $k$ | Timestep in predicted plan |
| $\Delta t$ | Time interval between discrete decision moments. $t = k\Delta t$ |
| $T$ | The transpose of a vector is indicated by superscript $T$ |
| Sets of nodes | |
| $\mathcal{N}$ | Set of all locations in the network, called network nodes |
| $\mathcal{D}$ | Set of virtual demand nodes. A virtual demand node is adjacent to the network node it shares ID with |
| $\mathcal{W}$ | Set consisting of the two directions of the waterway connection |
| $\mathcal{R}$ | Set of both directions of all roads in the network |
| Model parameters | |
| $n_c$ | Number of different container commodities. All containers of the same commodity have the same destination |
| $\bar{d}_i(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Demand realised at node $i$ at timestep $k$. Each element in the vector is the number of containers of that commodity. Commodities with destination in $i$ are to be delivered, the other commodities are ready for pick up |
| $p_i^r \in \{0, 1\}^{1 \times n_c}$ | Mapping that describes which elements in the demand state vector that are containers released the node, i.e. $p_i^r \bar{d}_i(k)$ is the sum of containers that are released at node $i$ at time $k$ |
| $p_i^d \in \{0, 1\}^{1 \times n_c}$ | Mapping that describes which elements in the demand state vector that are due in the node, i.e. $p_i^d \bar{d}_i(k)$ is the sum of containers that are due at node $i$ at time $k$ |
| $\tau_{12}^b, \tau_{21}^b$ | Travel time measured in timesteps by barge including mooring, (un)loading etc. from node 1 to 2 and reverse |
| $c^b$ | Barge capacity |
| $\tau_{ij}^r$ | Travel time measured in timesteps by truck from node $i$ to node $j$. We do not differentiate between travelling empty or full and disregard loading and unloading time for trucks |
| | One truck can carry one container |
| $\bar{z}_i^d(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Number of containers available for pickup or to be delivered at timestep $k$ in node $i$. Each element in the vector is the number of containers of that commodity. Commodities with destination in $i$ are to be delivered, the other commodities are ready for pick up |
| $\bar{z}_i^b(k) \in \{0, 1\}$ | $\bar{z}_i^b(k) = 1$ if the barge is docked (after unloading, before loading) at the quay in node $i$ at timestep $k$ |
| $\bar{z}_i^v(k) \in \mathbb{R}$ | Number of trucks parked at node $i$ at time $k$ |
| $\bar{z}_i^c(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Number of containers that are stacked at node $i$ at time $k$. Each element in the vector is the number of containers of that commodity |
| $\bar{x}^t(k)$ | State vector for the truck operator, containing the current state measurements and future arrivals of already departed trucks and containers |
| $\bar{x}^b(k)$ | State vector for the barge operator, containing the current state measurements and future arrivals of already departed barges |

**Table 4** continued

| Decision variables | |
| --- | --- |
| $\bar{y}_i(k) \in \{0, 1\}$ | $\bar{y}_i(k) = 1$ if the barge departs from node $i$ at timestep $k$ |
| $\bar{u}_{ij}^b(k) \in \mathbb{R}^{n_c}$ | Number of containers to load at timestep $k$ to the barge at location $i$ going to location $j$. Each element in the vector is the number of containers of that commodity |
| $\bar{u}_{ij}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Number of containers to load to trucks departing timestep $k$ from location $i$ going to location $j$. Each element in the vector is the number of containers of that commodity |
| $\bar{u}_{di}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Number of new containers entering the network node $i$ at timestep $k$. Always zero in elements corresponding to containers commodities with destination at the same node $i$ |
| $\bar{u}_{id}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Number of containers being delivered at node $i$ at timestep $k$. Always zero in elements corresponding to containers commodities with destination elsewhere |
| $\bar{v}_{ij}(k) \in \mathbb{R}$ | Number of trucks departing from $i$ on the road to $j$ at time $k$ |

Parameters and variables used during real-time planning

Notice: these predictions are updated at every decision moment. The full notation would include the time the prediction was made, e.g. $x(k|t)$ is the full state of the system at timestep $k$ in a prediction made at time $t$, but to simplify the reading of equations we throughout the paper use $x(k)$ to describe the full state of the system in the current prediction. This table only shows the simplified notation

| | |
| --- | --- |
| $T_p$ | Prediction horizon. Number of timesteps for which the receding horizon plan is made |
| $d_i(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Currently expected demand at node $i$ at timestep $k$. Each element in the vector is the number of containers of that commodity. Commodities with destination in $i$ are to be delivered, the other commodities are ready for pick up |
| $z_i^d(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Predicted number of containers available for pickup or to be delivered at timestep $k$ in node $i$. Each element in the vector is the number of containers of that commodity. Commodities with destination in $i$ are to be delivered, the other commodities are ready for pick up |
| $z_i^b(k) \in \{0, 1\}$ | Predicted availability of barge $z_i^b(k) = 1$ if the barge is ready for (un)loading at node $i$ at timestep $k$ |
| $z_i^v(k) \in \mathbb{R}$ | Predicted number of trucks parked at node $i$ at time $k$ |
| $z_i^c(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Number of containers that are stacked at node $i$ at time $k$. Each element in the vector is the number of containers of that commodity |
| $x^t(k)$ | Predicted state vector for the truck, containing the current state measurements and future arrivals of already departed trucks and containers |
| $x^b(k)$ | Predicted state vector for the barge operator, containing the current state measurements and future arrivals of already departed barges |
| $w_d$ | Cost per timestep delay per container |
| $w_i^b$ | Cost of sailing the empty barge when it departs from node $i$ |
| $w_{ij}^v \in \mathbb{R}$ | Cost of driving a truck (empty or full) from $i$ to $j$ |
| $w_{ij}^l \in \mathbb{R}_{\geq 0}^{1 \times n_c}$ | Cost of loading, unloading and handling one container when it is transported by barge from node $i$ to $j$. Each element correspond to the specific cost for that commodity |
| $J^b(k)$ | Total cost for barge operator related to departures at timestep $k$ |
| $J^t(k)$ | Total cost for truck operator for departures and unsatisfied demand at timestep $k$ |
| $y_i(k) \in \{0, 1\}$ | Predicted barge departures. $y_i(k) = 1$ if the barge departs from node $i$ at timestep $k$ |
| $u_{ij}^b(k) \in \mathbb{R}^{n_c}$ | Predictions loading of containers to the barge at timestep $k$ at location $i$ going to location $j$. Each element in the vector is the number of containers of that commodity |
| $u_{ij}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Predictions loading of containers to trucks departing timestep $k$ from location $i$ going to location $j$. Each element in the vector is the number of containers of that commodity |

**Table 4** continued

| | |
|---|---|
| $u_{di}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Predicted number of new containers entering the network node $i$ at timestep $k$. Always zero in elements corresponding to containers commodities with destination at the same node $i$ |
| $u_{id}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ | Predicted number of containers to be delivered at node $i$ at timestep $k$. Always zero in elements corresponding to containers commodities with destination elsewhere |
| $v_{ij}(k) \in \mathbb{R}$ | Predicted number of trucks departing from $i$ on the road to $j$ at time $k$ |
| Departure learning sets, parameters and variables | |
| $b_k \in \{0, 1\}$ | $b_k = 1$ if there is a barge departure (regardless from which terminal) |
| $e$ | Event: Plan of barge departures without location information, i.e. $e = [b_k, \ldots, b_{k+T_p-1}]$. Figure 3 shows an example |
| $\mathcal{E}(k)$ | Set of events that are feasible (i.e. respect barge travel time and current location of barge) at time $k$ |
| $e^\infty$ | Event expanded to an infinite timespan, i.e. $e^\infty = \begin{bmatrix} \mathbf{0}_{1:k} & e & \mathbf{0}_{k+T_p:\infty} \end{bmatrix}$. Figure 4 shows an example |
| $\mathcal{N}_{e_1^\infty}(k)$ | Set of neighbouring events. Figure 4 shows an example. If two events are identical except for two subsequent elements, the events are said to be neighbours, i.e. for an event $e_1 = [b_k^1, \ldots, b_{k+T_p-2}^1] \in \mathcal{E}(k)$, the set of neighbouring events is $\mathcal{N}_{e_1^\infty}(k) = \left\{ e = [b_k^2, \ldots, b_{k+T_p-2}^2] \in \mathcal{E}(k) \mid b_i^2 = b_i^1 \; \forall i \setminus \{i = \{j, j+1\} \text{ for exactly one } j \in \{k, \ldots, k+T_p-2\} \text{ and } b_j^2 = b_{j+1}^1, \; b_{j+1}^2 = b_j^1 \right\} \setminus \left\{ e_1 \right\}$ |
| $\mathcal{I}(k)$ | Set of Candidate events |
| $n$ | Cardinality of $\mathcal{I}(k)$, i.e. the number of schedules that is communicated and that the truck operator must evaluate at each decision moment |
| $\tilde{F}_{e^\infty}(k) \in \mathbb{R}$ | Expected fitness of event $e$ given what is known at time $k$. Describes the expected cost of implementing the schedule corresponding to the event $e$ |
| $\tilde{s}_{e^\infty}(k) \in \mathbb{R}$ | Uncertainty function of event $e$ given what is known at time $k$. Indicates how certain the expected fitness is. Lower value indicates confidence in expected fitness is higher |
| $\alpha \in [0, 1]$ | First learning parameter: used to update the expected fitness. Higher value puts emphasis previous expectations for the same events. Lower value shifts emphasis to expectations for neighbouring events |
| $\beta$ | Second learning parameter: controls the speed at which information from previous timesteps become uncertain |
| $J_{e^\infty}(k)$ | The optimal predicted cost for the truck operator if the barge follows the schedule corresponding to the event $e$ from timestep $k$ to $k+T_p-1$ |
| $\tilde{F}_{e^\infty}(k)$ | Fitness of event $e$. The optimal expected cost for both the barge and truck operator for the time period from timestep $k$ to $k+T_p$ if event $e$ |
| $\tilde{F}_{init}$ | Initialization of expected fitness, should be higher than the expected maximum fitness |
| $s_{new}$ | Initialization of the uncertainty function for new events |

# References

Baptista, R., & Poloczek, M. (2018). Bayesian optimization of combinatorial structures. In *International conference on machine learning* (pp. 462–471). PMLR.

Behdani, B., Fan, Y., Wiegmans, B., & Zuidwijk, R. (2016). Multimodal schedule design for synchromodal freight transport systems. *European Journal of Transport and Infrastructure Research, 16*(3), 424–444.

Boros, E., Lei, L., Zhao, Y., & Zhong, H. (2008). Scheduling vessels and container-yard operations with conflicting objectives. *Annals of Operations Research, 161*, 149–170.

Cruijssen, F., Dullaert, W., & Fleuren, H. (2007). Horizontal cooperation in transport and logistics: A literature review. *Transportation Journal, 46*(3), 22–39.

Demir, E., Burgholzer, W., Hrušovskỳ, M., Arikan, E., Jammernegg, W., & Van Woensel, T. (2016). A green intermodal service network design problem with travel time uncertainty. *Transportation Research Part B: Methodological, 93*, 789–807.

Di Febbraro, A., Sacco, N., & Saeednia, M. (2016). An agent-based framework for cooperative planning of intermodal freight transport chains. *Transportation Research Part C: Emerging Technologies, 64*, 72–85.

Eurostat. (2020). *Databse: Annual road freight transport vehicle movements, loaded and empty, by reporting country*. https://data.europa.eu/euodp/en/data/dataset/oXV2zjVuHDQNMsQX8vsw.

Frazier, P. I. (2018). *A tutorial on Bayesian optimization*.

Gansterer, M., & Hartl, R. F. (2018). Collaborative vehicle routing: A survey. *European Journal of Operational Research, 268*(1), 1–12.

Gansterer, M., Hartl, R. F., & Savelsbergh, M. (2020). The value of information in auction-based carrier collaborations. *International Journal of Production Economics, 221*, 107485. ISSN 0925-5273.

Giusti, R., Manerba, D., Bruno, G., & Tadei, R. (2019). Synchromodal logistics: An overview of critical success factors, enabling technologies, and open research issues. *Transportation Research Part E: Logistics and Transportation Review, 129*, 92–110.

Huang, Y., Zhou, Q., Xiong, X., & Zhao, J. (2021). A cooperative intermodal transportation network flow control method based on model predictive control. *Journal of Advanced Transportation, 6*, 66.

Juan, A. A., Keenan, P., Martí, R., McGarraghy, S., Panadero, J., Carroll, P., & Oliva, D. (2021). A review of the role of heuristics in stochastic optimisation: From metaheuristics to learnheuristics. *Annals of Operations Research, 66*, 1–31.

Kandasamy, K., Krishnamurthy, A., Schneider, J., & Póczos, B. (2018). Parallelised Bayesian optimisation via thompson sampling. In *International conference on artificial intelligence and statistics* (pp. 133–142). PMLR.

Karam, A., Hussein, M., & Reinau, K. H. (2021). Analysis of the barriers to implementing horizontal collaborative transport using a hybrid fuzzy delphi-ahp approach. *Journal of Cleaner Production, 321*, 128943.

Larsen, R. B., Atasoy, B., & Negenborn, R. R. (2020). Learning-based co-planning for improved container, barge and truck routing. In E. Lalla-Ruiz, M. Mes, & S. Voß, S. (Eds.), Proceedings of the international conference on computational logistics (pp. 476–491).

Larsen, R. B., Atasoy, B., & Negenborn, R. R. (2021a). Model predictive control for simultaneous planning of container and vehicle routes. *European Journal of Control,57*, 273–283.

Larsen, R. B., Baksteen, R., Atasoy, B., & Negenborn, R. R. (2021b). Secure multi-party co-planning of barge departures. *IFAC-PapersOnLine,54*(2), 335–341. IFAC symposium on control in transportation systems.

Larsen, R. B., Sprokkereef, J. M., Atasoy, B., & Negenborn, R. R. (2021c). Integrated mode choice and vehicle routing for container transport. In *Proceedings of the international intelligent transportation systems conference* (pp. 3348–3353).

Li, L., Negenborn, R. R., & De Schutter, B. (2015). Intermodal freight transport planning—A receding horizon control approach. *Transportation Research Part C: Emerging Technologies, 60*, 77–95.

Li, L., Negenborn, R. R., & De Schutter, B. (2017). Distributed model predictive control for cooperative synchromodal freight transport. *Transportation Research Part E: Logistics and Transportation Review, 105*, 240–260.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in matlab. In *Proceedings of international symposium on computer aided control system design* (pp. 284–289).

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica, 36*(6), 789–814.

Nabais, J. L., Negenborn, R. R., & Botto, M. A. (2013). Model predictive control for a sustainable transport modal split at intermodal container hubs. In *Proceedings of the international conference on networking, sensing and control* (pp. 591–596).

Očenášek, J., & Schwarz, J. (2000). The parallel Bayesian optimization algorithm. In *The state of the art in computational intelligence* (pp. 61–67). Springer.

Oh, C., Tomczak, J., Gavves, E., & Welling, M. (2019). Combinatorial Bayesian optimization using the graph cartesian product. *Advances in Neural Information Processing Systems, 32*, 66.

Pérez Rivera, A. E., & Mes, M. R. K. (2019). Integrated scheduling of drayage and long-haul operations in synchromodal transport. *Flexible Services and Manufacturing Journal, 31*(3), 763–806.

Poloczek, M., Wang, J., & Frazier, P. I. (2016). Warm starting Bayesian optimization. In *2016 Winter simulation conference (WSC)* (pp. 770–781). IEEE.

Poo, M.C.-P., & Yip, T. L. (2019). An optimization model for container inventory management. *Annals of Operations Research, 273*(1), 433–453.

Psaraftis, H. N., Wen, M., & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks, 67*(1), 3–31.

Qu, W., Rezaei, J., Maknoon, Y., & Tavasszy, L. (2019). Hinterland freight transportation replanning model under the framework of synchromodality. *Transportation Research Part E: Logistics and Transportation Review, 131*, 308–328.

Resat, H. G., & Turkay, M. (2019). A discrete-continuous optimization approach for the design and operation of synchromodal transportation networks. *Computers & Industrial Engineering, 130*, 512–525.

Rivera, A. E. P., & Mes, M. R. K. (2022). Anticipatory scheduling of synchromodal transport using approximate dynamic programming. *Annals of Operations Research, 66*, 1–35.

Sager, S., Bock, H. G., & Diehl, M. (2012). The integer approximation error in mixed-integer optimal control. *Mathematical Programming, 133*(1), 1–23.

SteadieSeifi, M., Dellaert, N. P., Nuijten, W., van Woensel, T., & Raoufi, R. (2014). Multimodal freight transportation planning: A literature review. *European Journal of Operational Research, 233*(1), 1–15.

van Essen, H., van Wijngaarden, L., Schroten, A., de Bruyn, S., Sutter, D., Bieler, C., Maffii, S., Brambilla, M., Fiorello, D., Fermi, F., Parolin, R., & El Beyrouty, K. (2019). *Handbook on the external costs of transport*. CE Delft. www.cedelft.eu

van Riessen, B., Negenborn, R. R., & Dekker, R. (2015). Synchromodal container transportation: An overview of current topics and research opportunities. In F. Corman, S. Voß, & R. R. Negenborn (Eds.), Proceedings of the international conference on computational logistics (pp. 386–397). Springer.

van Riessen, B., Negenborn, R. R., Dekker, R., & Lodewijks, G. (2015). Service network design for an intermodal container network with flexible transit times and the possibility of using subcontracted transport. *International Journal of Shipping and Transport Logistics, 7*(4), 457–478.

Xu, S. X., Huang, G. Q., & Cheng, M. (2017). Truthful, budget-balanced bundle double auctions for carrier collaboration. *Transportation Science, 51*(4), 1365–1386.

Xu, Y., Cao, C., Jia, B., & Zang, G. (2015). Model and algorithm for container allocation problem with random freight demands in synchromodal transportation. *Mathematical Problems in Engineering*, Article ID 986152.

Yang, R., Lee, C.-Y., Liu, Q., & Zheng, S. (2019). A carrier-shipper contract under asymmetric information in the ocean transport industry. *Annals of Operations Research, 273*(1–2), 377–408.