Learning-Based Co-planning for Improved Container, Barge and Truck Routing

Larsen, Rie B.; Atasoy, Bilge; Negenborn, Rudy R.

DOI
10.1007/978-3-030-59747-4_31

Publication date
2020

Document Version
Final published version

Published in
Computational Logistics

Learning-Based Co-planning for Improved Container, Barge and Truck Routing

Rie B. Larsen(✉), Bilge Atasoy, and Rudy R. Negenborn

Department of Maritime and Transport Technology,
Delft University of Technology, Delft, The Netherlands
{r.b.larsen,b.atasoy,r.r.negenborn}@tudelft.nl

Abstract. When barges are scheduled before the demand for container transport is known, the scheduled departures may match poorly with the realised demands' due dates and with the truck utilization. Synchromodal transport enables simultaneous planning of container, truck and barge routes at the operational level. Often these decisions are taken by multiple stakeholders who wants cooperation, but are reluctant to share information. We propose a novel co-planning framework, called departure learning, where a barge operator learns what departure times perform better based on indications from the other operator. The framework is suitable for real time implementation and thus handles uncertainties by replanning. Simulated experiment results show that co-planning has a big impact on vehicle utilization and that departure learning is a promising tool for co-planning.

Keywords: Cooperative planning · Synchromodal transport · Vehicle utilization

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. One of the main challenges of humanity at the moment is the climate changes. One way of alleviating our negative impact on the environment is to increase the efficiency of our activities. The transport sector is a large contributor of CO2 emissions and has a low efficiency, with e.g. trucks driving empty 26% of the kilometres they drive in the Netherlands [5]. CO2 emission is however not the only negative impact of freight transport. The report [4] 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

This research is supported by the project "Complexity Methods for Predictive Synchromodality" (project 439.16.120) of the Netherlands Organisation for Scientific Research (NWO).

E. Lalla-Ruiz et al. (Eds.): ICCL 2020, LNCS 12433, pp. 476–491, 2020.
https://doi.org/10.1007/978-3-030-59747-4_31

in this order [4, Fig. 16]. It is therefore desirable not only to improve the vehicle utilization, and hence efficiency, of truck transport, but also the utilization across transport modes.

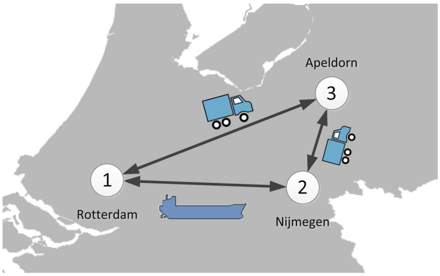

Fig. 1. Small Dutch transport network used as example in this paper.

Synchromodal transport uses a-modal bookings and change acceptance to enable transport providers to optimize plans in accordance with the realisation of uncertainties. In the traditional transport literature, decisions are divided into strategic, tactical and operational levels [15]. 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 regards 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., [9,11] and [7].

Barge schedules are typically decided at the tactical level based on estimated demand [3]. 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 [16], others adjust predefined departure times after the demand realization [1] or cancel unprofitable departures [18]. Truck routing is typically decided at the operational level based on pick up and delivery locations and times of the goods [12]. In [6] and [7] 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 [13] shows 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 co-planning. Co-planning can involve both information sharing and loss of autonomy. Many companies are interested in the benefits of cooperation [2], but participate reluctantly due to these implications. Co-planning schemes can be constructed such that missing information or sudden changes in the willingness to follow the scheme can damage the other participating parties. Cooperation schemes vary from auctions [17] to distributed optimization [8]. 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.

In this paper, we show the impact of co-planning and develop a method, called *departure learning* (DL), for real time co-planning between a barge and a truck operator. DL can be generalized to multiple truck operators. The method requires communication of a number of schedules and indications of the corresponding performances between the barge operator and the truck operator. The performances can be communicated as ratios to mask the real numbers behind. It is assumed no party seeks to exploit the framework, but it does not severely damage cooperating parties if one party acts autonomously. The framework is based on Model Predictive Control (MPC) and uses ideas from Bayesian optimization to learn good departure times through continuous communication.

2 Real Time Co-planning in Synchromodal Networks

The synchromodal container transport networks we consider in this paper are described by graphs $\mathcal{G}(\mathcal{N},\mathcal{A})$ where the nodes $\mathcal{N}$ are terminals and the arches $\mathcal{A}$ are connecting infrastructure. One arc corresponds to inland waterways and can thus only be used by barges and others are roads used by trucks. The set of road-arcs is denoted $\mathcal{R}$ and the two directions of the waterway comprise the set $\mathcal{W}$. Two nodes can be connected by both types of arcs. 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 only the truck operators have contact to clients and therefore the barge operator receives the demand only through truck operators. Figure 1 shows the transport network used as example in this paper.

It is assumed that barge and truck operators want to collaborate to decrease the total cost of transport and they have agreed how to share the profit. The truck operators are not willing to share information on release time, due date and quantity of their transport orders, but are willing to indicate how costly different barge schedules will be to them. The barge operator has the final authority to decide the schedule but adjusts it based on the feedback from the operators. The feedback from the operators are collectively considered by a weighted sum. For clarity, the DL is therefore presented for networks with one barge operator and one truck operator. It is furthermore assumed that both parties commit fully to the proposed framework. It is however worth noticing that agreement does not need to be reached since the barge operator has the final saying over the schedule and the truck company has authority to route the containers and is the only party who knows destinations and due dates.

Since the responsibilities of the barge and the truck operator are divided, the following description of the synchromodal transport network is also divided. Before this description, the real time aspect of the network and DL are discussed.

2.1 Real Time Aspect

Uncertainties are very common in the transport sector. To address them, DL is based on MPC, where time t is divided into timesteps k with Δt timeunits in between such that $t = k\Delta t$. The optimal decisions are found by optimizing the system performance over a prediction horizon T_p. Only the decisions regarding the current timestep k are implemented and at the next time $k+1$ the process is repeated. MPC thus react to changes in the system or predictions every Δt time units and considers the period k to $k + T_p$ when it takes decisions. In the co-planning problem, 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. Using MPC for problems that requires frequent updates, i.e. low Δt, and a long prediction horizon T_p requires fast optimization of the model. We therefore formulate the truck and container routing problem with continuous variables. Frequent updates can ensure sufficient precision when the continuous optimal decisions are rounded to integer variables [14]. The barge capacity is much larger than that of trucks and they are thus described by discrete variables.

2.2 Barge Operator

The barge operator is responsible for the barge schedule. It is assumed the synchromodal transport network only has two barge terminals: nodes 1 and 2. The travel time from node 1 to 2, τ_{12}^b, and the return, τ_{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.

Two binary variables $y_1(k)$ and $y_2(k)$ are used to describe the departures of the barge at time step k from node 1 and 2 respectively. The dynamics of the barges can thus be described as

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

where $\bar{z}_i^b(k) \in \{0, 1\}$ is the number of barges at the quay of node i at time k. The superscript b is used to indicate the variable regards the barge and the bar on top, that it is the realized value. The barge operator knows the position of the barge at all time, since he has access to the barge state

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

The main cost for the barge operator is sailing the barge, since owning the equipment and hiring people are out of scope of this problem. There are additional costs involved with transporting containers on the barge, e.g., crane movements and increased weight of the barge. The total cost is thus $w_1^b\bar{y}_1(k) + w_2^b\bar{y}_2(k) + w_{12}^l\bar{u}_{12}^b(k) + w_{12}^l\bar{u}_{12}^b(k)\ \forall\, k$, where w_i^b is the cost of sailing an empty barge from i to j, $\{i,j\} = \{1,2\}$, and $w_{ij}^l \in \mathbb{R}_{\geq 0}^{1\times n_c}$ is the cost of transporting one additional container with the barge from i to j. $u_{ij}^b(k) \in \mathbb{R}_{\geq 0}^{1\times n_c}$ is a vector with the number of containers of each commodity that is transported from i to j by barge departing at time k. n_c is the number of commodities. Since the barge and truck operators cooperate fully and share the profit after the transport has been performed, the cost per container is considered by the truck operator. The private cost for the bare operator is

$$J^b(k) = w_1^b\bar{y}_1(k) + w_2^b\bar{y}_2(k) \quad \forall\, k. \tag{2}$$

2.3 Truck Operator

The truck operator is responsible for choosing which modes and routes each container is transported by and for deciding the truck routes. The model used to describe this simultaneous planning problem is a simplification of on the method presented by us in [7]. The full model can be used with DL, but as it ads complexity to the description, the simplified model is used here. The key assumptions of the truck operator problem are:

- Any node in the network can be the origin and destination of transport demand, if it is defined as such, hence both import and export 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 continuous variable, commodity flows. This simplifies the model and captures the desired level of accuracy ([11] and [14]).
- Trucks are also modelled as continuous variable flows.
- The number of trucks is finite and each truck can transport one container.
- The barge has invariant, finite capacity. Other capacities and (un)loading rates are considered sufficient.
- Terminal operating hours, drivers resting hours, etc., are not considered.

All containers with the same destination are modelled as one commodity. Since the containers are described as flows, one container of a certain commodity can replace another. This assumption is also used in, e.g., [9]. More commodities can have the same destination, which allow us to distinguish e.g. different container sizes. In [11] a commodity is defined for each due date to ensure all containers arrive on time. We define virtual demand nodes adjacent to the nodes where containers can have origin or destination. The set of virtual nodes is denoted by $\mathcal{D}$. The dynamics of the virtual demand nodes are

$$\bar{z}_i^d(k+1) = \bar{z}_i^d(k) - \bar{u}_{di}(k) - \bar{u}_{id}(k) + \bar{d}_i(k) \quad \forall i \in \mathcal{D},\ \forall\, k, \tag{3}$$

where $\bar{d}_i(k) \in \mathbb{R}_{\geq 0}^{n_c}$ is the realised new demand of each commodity at time k. Notice that all values are positive, so whether the demand indicates container releases or expected arrivals depends on the commodity, i.e. the element in the vector. 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 released 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:

$$p_i^r \bar{u}_{id}(k) = 0 \quad \forall\, i \in \mathcal{D},\ \forall\, k \tag{4}$$

$$p_i^d \bar{u}_{di}(k) = 0 \quad \forall\, i \in \mathcal{D},\ \forall\, k \tag{5}$$

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 has 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

$$\begin{aligned}\bar{z}_i^c(k+1) = \bar{z}_i^c(k) &+ \sum_{j\in\mathcal{D}_i} \left(\bar{u}_{di}(k) - \bar{u}_{id}(k)\right) + \sum_{j\in\mathcal{W}_i} \left(\bar{u}_{ji}^b(k-\tau_{ji}^b) - \bar{u}_{ij}^b(k)\right) \\ &+ \sum_{j\in\mathcal{R}_i} \left(\bar{u}_{ji}(k-\tau_{ji}^r) - \bar{u}_{ij}(k)\right) \quad \forall\, i \in \mathcal{N}, \forall\, k.\end{aligned} \tag{6}$$

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. The superscript c indicates that the variable regards containers. $u_{ij}(k) \in \mathbb{R}_{\geq 0}^{n_c}$ has the same structure and is for the containers transported from i to j by road at time k. The travel takes τ_{ij} timesteps. The barge has a capacity of c^b and only carries containers that were ready for loading at the departure time. Hence

$$\mathbf{1}_{n_c} \bar{u}_{ij}^b(k) \leq c^b \bar{y}_i(k) \quad \forall\, <i,j> \in \mathcal{W},\ \forall\, k, \tag{7}$$

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

$$\bar{z}_i^v(k+1) = \bar{z}_i^v(k) + \sum_{j\in\mathcal{R}_\rangle} \bar{v}_{ji}(k-\tau_{ji}^r) - \bar{v}_{ij}(k) \quad \forall\, i \in \mathcal{N},\ \forall\, k, \tag{8}$$

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

$$\mathbf{1}_{n_c} u_{ij}(k) \leq v_{ij}(k) \quad \forall\, j \in \mathcal{R}_i,\ \forall\, i \in \mathcal{N},\ \forall\, k. \tag{9}$$

Since the truck and the barge operators cooperate fully and share the profit after the transport has been performed, the truck operator considers all costs that are directly related to his decisions, namely

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

where $w_{ij}^v \in \mathcal{R}$ is the cost of driving a truck from i to j and w_d is the cost per timestep delay per container. The truck operator has always access to the state

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

3 Departure Learning

The cooperative planning between the barge and truck operator is based on exchange of information and commitment to find the solution that is cheapest for both parties. The truck operator is not willing to share information about specific containers and the barge operator wants autonomy over the schedule. We propose the novel method *departure dearning* (DL), where at each timestep, the barge operator sends a set $\mathcal{I}(k)$ of barge schedules to the truck operator. The truck operator hereafter computes the transport cost over the prediction horizon for each of the schedules. The costs are send to the barge operator, possibly after scaling to hide the exact information.

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 DL is shown in Fig. 2. In the following, it is described how the barge operator learns good schedules, and how the truck operator estimates the performance of a schedule.

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 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

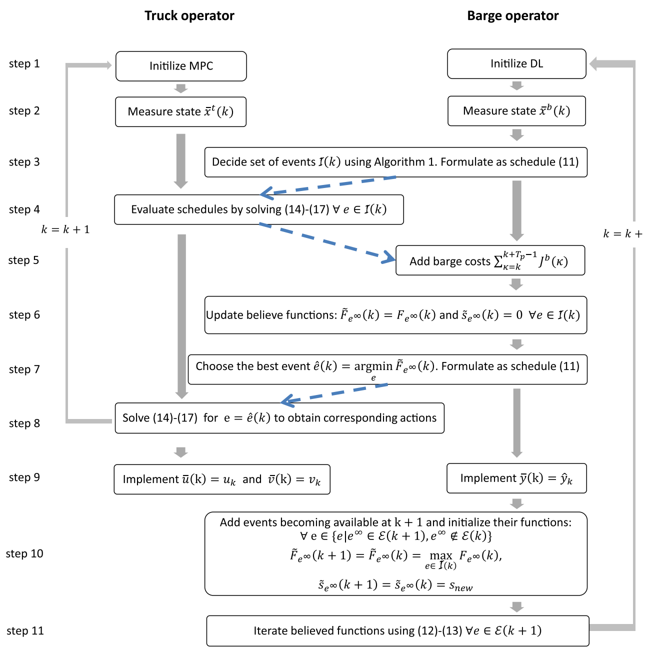

Fig. 2. Actions of DL. Blue, dashed arrows indicate communication. (Color figure online)

$$\begin{array}{|ll|}\hline [y_1(k) \cdots y_1(k+T_p-1)] & = [\ 0\ 0\ 1\ 0\ 0\ 0\ ] \\ [y_2(k) \cdots y_2(k+T_p-1)] & = [\ 1\ 0\ 0\ 0\ 1\ 0\ ] \\ e & = [\ 1\ 0\ 1\ 0\ 1\ 0\ ] \\ \hline \end{array}$$

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

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 and this relation:

$$y_i(k+\gamma) \leq z_i^b(k) + \sum_{\kappa=\tau_{ji}^b-\gamma}^{\tau_{ji}^b} y_j(k-\kappa) \quad \forall \gamma \leq \tau_{ji}, \forall < i,j > \in \mathcal{W}. \tag{11}$$

Each element of the event is a binary variable denoted by b_k. An event is thus $e = [b_k, ..., b_{k+T_p-1}]$ where each element is a specific realizations of $b_k \in \{0,1\}, ..., b_{k+T_p-1} \in \{0,1\}$. 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 are as such identical. e^∞ denotes an event over the infinite timespan and is defined as $e^{\infty} = \left[\mathbf{0}_{1:k}\ e\ \mathbf{0}_{k+T_p:\infty}\right]$, 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 $e_1 = [b_k^1, ..., b_{k+T_p-2}^1] \in \mathcal{E}(k)$, the set of neighbouring events is $\mathcal{N}_{e_1^\infty}(k) = \Big\{ e = [b_k^2, ..., b_{k+T_p-2}^2] \in \mathcal{E}(k) \,|\, b_i^2 = b_i^1\ \forall\, i \setminus \{i = j+1\}$ for one $j \in \{1, ..., T_p - 2\}$ and $b_j^2 = b_{j+1}^1,\, b_{j+1}^2 = b_j^1 \Big\} \setminus \Big\{ e_1 \Big\}$. 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_∞ 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_∞ and $\mathcal{N}_{e^\infty}(k)$ are exemplified in Fig. 4.

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

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

It is expected that the performance indicator for events that share e^∞ evolve slowly over time, and that the performances of neighbouring events are related. The barge operator's estimate of the performance is called the event's expected fitness and is denoted by $\tilde{F}_{e^\infty}(k)$. To indicate how certain this estimate is, an uncertainty function $\tilde{s}_{e^\infty}(k)$ is used. $\tilde{s}_{e^\infty}(k)$ decreases when an event corresponding to e^∞ or its neighbours are evaluated and increases slowly over k. If the barge operator has received the performance indicator for an event e, we say event e has been evaluated. 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 DL, 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 performance indicators from the truck operator, the expected fitness of the evaluated events are updated and their uncertainty values are set to zero. Some events will be feasible at the next time $k+1$ which were not feasible at time k. These events are initialized with the

Algorithm 1. The strategy used to decide $\mathcal{I}(k)$

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: $\quad e_{new} = \arg\min_{e \in \mathcal{E}(k) \setminus \mathcal{I}(k)} \tilde{F}_{e^\infty}(k) + \tilde{s}_{e^\infty}(k)$
6: $\quad \mathcal{I}(k) = \mathcal{I}(k) \cup e_{new}$
7: $\quad e_{new} = \arg\min_{e \in \mathcal{E}(k) \setminus \mathcal{I}(k)} \tilde{F}_{e^\infty}(k)$
8: $\quad \mathcal{I}(k) = \mathcal{I}(k) \cup e_{new}$
9: $\quad e_{new} = \arg\max_{e \in \mathcal{E}(k) \setminus \mathcal{I}(k)} \tilde{s}_{e^\infty}(k)$
10: $\quad \mathcal{I}(k) = \mathcal{I}(k) \cup e_{new}$
11: $\quad e_{new} = \arg\min_{e \in \mathcal{E}(k) \setminus \mathcal{I}(k)} \tilde{F}_{e^\infty}(k) - \tilde{s}_{e^\infty}(k)$
12: $\quad \mathcal{I}(k) = \mathcal{I}(k) \cup e_{new}$
13: $\quad$ **for** $j \leftarrow 1$ to 2 **do**
14: $\qquad e_{new} = \text{rand}\,(e \in \mathcal{E}(k) \setminus \mathcal{I}(k))$
15: $\qquad \mathcal{I}(k) = \mathcal{I}(k) \cup e_{new}$
16: $\quad$ **end for**
17: **end for**
18: **for** $i \leftarrow floor(n/6)6$ to n **do**
19: $\quad e_{new} = \text{rand}\,(e \in \mathcal{E}(k) \setminus \mathcal{I}(k))$
20: $\quad \mathcal{I}(k) = \mathcal{I}(k) \cup e_{new}$
21: **end for**

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:

$$\tilde{F}_{e^\infty}(k+1) = \alpha \tilde{F}_{e^\infty}(k) + \frac{1-\alpha}{|\mathcal{N}_{e^\infty}(k)|} \sum_{i \in \mathcal{N}_{e^\infty}(k) \cup \mathcal{N}_{e^\infty}(k+1)} \tilde{F}_i(k) \tag{12}$$

$$\tilde{S}_{e^\infty}(k+1) = (\alpha+\beta) \tilde{S}_{e^\infty}(k) + \frac{1-\alpha}{|\mathcal{N}_{e^\infty}(k)|} \sum_{i \in \mathcal{N}_{e^\infty}(k) \cup \mathcal{N}_{e^\infty}(k+1)} \tilde{S}_i(k) \tag{13}$$

The learning parameter α balances the emphasis laid on each events' previous value and on neighbouring events' values and t he factor β controls the speed at which information from previous timesteps become uncertain. To initialize DL 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 at step 10 in Algorithm 1.

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 optimization problem (14)–(17), initiated from the current state for the given schedule.

$$\tilde{F}_{e^{\infty}}(k) = \min \sum_{\kappa=k}^{k+T_p-1} J^t(\kappa) \tag{14}$$

$$\text{s.t. } x^t(k) = \bar{x}^t(k) \tag{15}$$

$$\{< y_1(\kappa), y_2(\kappa) > | \kappa \in \{k, ..., k+T_p-1\}\} = e \tag{16}$$

$$(3)-(9) \quad \forall \kappa \in \{k, ..., k+T_p-1\} \tag{17}$$

$$\min \sum_{\kappa=k}^{k+T_p-1} J^t(\kappa) + J^b(\kappa) \tag{18}$$

$$\text{s.t. } x^t(k) = \bar{x}^t(k) \tag{19}$$

$$x^b(k) = \bar{x}^b(k) \tag{20}$$

$$(1),\ (3)-(9) \qquad \forall \kappa \in \{k, ..., k+T_p-1\} \tag{21}$$

$$y_1(\kappa) \in \{0,1\},\ y_1(\kappa) \in \{0,1\} \quad \forall \kappa \in \{k, ..., k+T_p-1\} \tag{22}$$

4 Simulation Experiments

To illustrate the impact of co-planning of barges, trucks and containers, two sets of simulated experiments are carried out. The first small scale experiment provides a better understanding of the possibilities for better utilization of the barge and trucks. The second experiment shows both the impact of co-planning and of using DL in a realistic scenario for the network in Fig. 1. The experiments are performed in Matlab formulated with Yalmip [10] and solved by Gurobi.

DL is benchmarked against co-planning based on centralised optimization and planning based on fixed schedules. The former benchmark is the best possible solution, henceforth called optimal co-planning, while the latter represents common practice. Both methods use MPC. Optimal co-planning solves the optimization problem (18)–(22) at each time k and when the schedules are fixed, the truck problem (14)–(17) is solved for that predefined schedule.

4.1 Experimental Setup

Three scenarios, on the network in Fig. 1 are used in the experiments. It is assumed Rotterdam and Apeldorn are origin and destination for demand and that all containers are of the same size, leading to $n_c = 2$ different commodities.

Realistic Scenario. The simulated experiments take place over 5 days and new decisions are taken every $\Delta t = 15$ min. It is assumed trucks drive 90 km/h and (un)loading a truck in Rotterdam takes 20 min, while it is 10 min in Nijmegen and Apeldorn. With these assumptions, the 140 km distance between Rotterdam and Apeldorn corresponds to 123 min traveltime, and the 55 km distance between Nijmegen and Apeldorn takes 56 min. The barge between Dordrecht and Nijmegen is in [13] reported to take 5 h including loading, so we assume the total travel time between Rotterdam and Nijmegen is 6 h. The cost of using the barge is in the same paper stated to be €60 per barge and €4.29 + €23.89 = €28.18 per container. The hourly rate of trucking is stated at €30.98 with a starting fee of €15. Using these values, the costs and travel times shown in Table 1 are computed. To ensure the consequences of a barge return trip are considered, the prediction horizon is chosen to be $T_p = 70$. The barge capacity is $c^b = 100$ and 36 trucks start at node 1.

The demand to be released at each virtual demand node is drawn from a uniform distribution between 0 and 3 at each timestep. The containers have a minimum lead time of 70 timesteps. The number of containers due at a virtual demand node is drawn at each timestep from a uniform destitution between zero and the number of containers that can have due date at this destination at this time. A total of 949 containers are released from node 1 and 984 from node 3. The same demand profile was used for all experiments.

DL is initialized with $\tilde{F}_{init} = 10000$ and $s_{new} = 5000$. The learning parameters are $\alpha = 0.8$ and $\beta = 0.1$. $n = 28$ potential schedules is communicated between operators at each timestep.

The benchmark method that builds on a fixed schedule uses regular barge departures. This barge leaves node 2 at time $k = 1$ and departs hereafter every $\tau_{12}^b + 2 = 26$ timesteps.

Vehicle-Centered Cost Scenario. To analyse the effect of changing the cost from the containers to the barge, we lowered the cost of transporting a container by €15 to $w_{12}^l = w_{21}^l =$ €128.18 per container. To make the total cost of sailing a full container the same, the cost of moving the barge was increased to $w_{12}^b = w_{21}^b =$ €210. The other parameters are the same as in the realistic scenario.

Table 1. Scenario parameters

$c^b = 100$	$\tau_{12}^b = \tau_{21}^b = 24$	$w_{12}^b = w_{21}^b = 60$	$w_{12}^l = w_{21}^l = 28.18\mathbf{1}_{n_c}$
$\bar{z}_1^v(0) = 36$	$\tau_{13} = \tau_{31} = 9$	$w_{13}^v = w_{31}^v = 73.19$	$w_d = 1000$
$\bar{z}_2^v(0) = 0$	$\tau_{23} = \tau_{32} = 4$	$w_{23}^v = w_{32}^v = 33.93$	

Realistic scenario

$c^b = 10$	$\tau_{12}^b = \tau_{21}^b = 2$	$w_{12}^b = w_{21}^b = 10$	$w_{12}^l = w_{21}^l = 28.18\mathbf{1}_{n_c}$
$\bar{z}_1^v(0) = 4$	$\tau_{13} = \tau_{31} = 2$	$w_{13}^v = w_{31}^v = 73.19$	$w_d = 1000$
$\bar{z}_2^v(0) = 0$	$\tau_{23} = \tau_{32} = 1$	$w_{23}^v = w_{32}^v = 33.93$	

Tractable scenario

Tractable Scenario. This scenario is a simple case to better illustrate the potential improvements. The total simulation time is 10 timesteps and the prediction horizon is $T_p = 7$. Due to the short prediction horizon, the barge operator has less time to learn what a good departure is. On the other hand, less events are feasible. Therefore it is chosen to communicate $n = 12$ schedules every timesteps. The learning parameters remain $\alpha = 0.8$ and $\beta = 0.1$. The capacities, travel times, and costs have been adjusted to the size of the problem and are as shown in Table 1. A total of 7 containers are to be transported. Four containers are released at virtual demand node 3 at time $k = 1$ with destination in virtual demand node 1. Two of them are due at $k = 4$ and two at $k = 6$. Three containers are released at $k = 3$ in virtual demand node 1 with due date at $k = 9$ in virtual demand node 3. The schedule used by the fixed-schedule benchmark method has a departure from node 2 at time $k = 5$ and a departure from node 1 at $k = 8$.

4.2 Results

For the tractable scenario, the realised movements of barges and trucks are shown in Fig. 5 for each of the three control strategies together with the demand, stacked containers and parked trucks at each node. The results show that when the barge schedule is optimized together with the truck and container routes, a better utilization of the barge is achieved. Optimal co-planning departs the barge twice and transport two containers each time. When DL is used, the barge departs once with two containers. The fixed schedule is not aligned with the transport need, and the barge is thus empty on both departures. The truck utilization is the same for optimal co-planning and fixed schedule with 8 out of 22 timesteps driven by trucks being without containers. For DL the utilization is lower with 11 out of 23 timesteps driven empty.

For both realistic sized scenarios, Table 2 shows key results provided by DL and the benchmark methods. Optimal co-planning obtains as expected the lowest cost, followed by DL. This shows that planning barge schedules before the realization of the demand leads to less flexibility and thus higher costs. The total number of containers transported by barge and the average number of containers per barge are significantly lower in the fixed schedule case than for optimal co-planning, despite the fixed schedule having more departures. The barge schedules achieved by DL have fewer departures and lower occupancy than by optimal co-planning. The number of containers per barge seems however to be very cost dependent with a higher occupancy when the cost is vehicle-centered, where DL achieves better occupancy than the fixed schedule. The number of trucks used by all three methods are comparable for both scenarios. Generally few trucks drive empty. This is likely because truck and container routes are optimized simultaneously in all methods. When the costs from [13] are used, optimal co-planning postpones demand satisfaction, resulting in 766 unsatisfied container-timesteps, corresponding to 191 container-hours. When the cost is vehicle-centered, optimal co-planning satisfy all demand in time. DL satisfies more demand than the fixed schedule case.

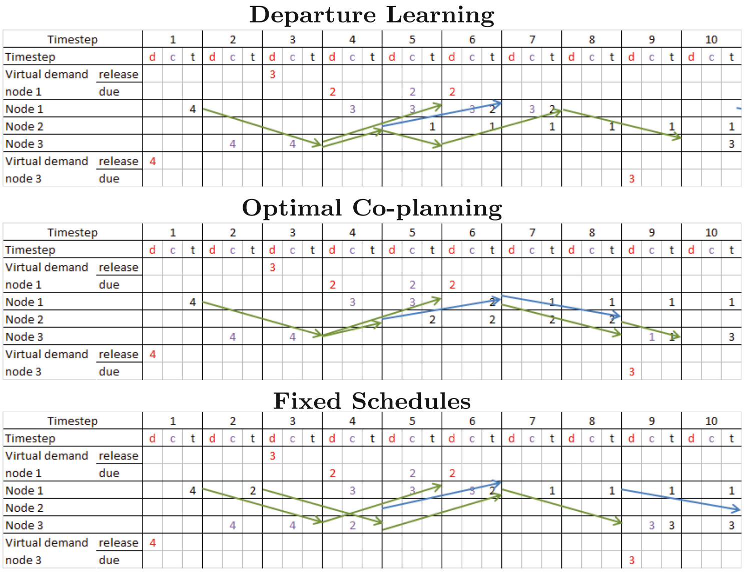

Fig. 5. Realised transport for the tractable scenario. Red numbers in column d are new demand and purple numbers in column c are containers stacked at the node. Black numbers in column t are vehicle parked at the node and green arrows indicate vehicle movements. Blue arrows are barge movements. (Color figure online)

Table 2. Results for the realistic scenario

Scenario	Costs from [13]			Vehicle-centered cost		
Planning method	DL	Optimal	Fixed	DL	Optimal	Fixed
Total cost (thousand €)	5982	5040	6148	5972	4928	6132
Unsatisfied demand	330.6	765.9	596.1	453.3	0	612.8
Barge departures	11	14	19	10	14	19
Containers transported by barge	207	501.4	385.1	278.1	551.7	396.0
Average barge occupancy (containers)	18.8	35.8	20.3	27.81	39.4	20.8
Truck departures	1746	1788.0	1745.8	1762	1813.8	1754
Empty truck departures	41.5	38.4	39	59.6	61.7	40

The results from both sets of experiments show that co-planning barge schedules with container and barge routes obtain better results in terms of cost. It furthermore increases the utilization rate of and number of containers transported by barge. Optimizing barge schedules together with container and truck routes achieves the best results. However, when one stakeholder does not have all information and authority over all decisions, DL provides a good alternative. The results show that DL is feasible in a realistic scenario and that it performs better than the fixed schedules case in terms of demand satisfaction and cost.

5 Conclusion

Planning barge schedules in real time together with truck and container routes results in a more efficient transport system. The presented results show that the total cost of transport is lowered and that optimal co-planning yields better barge utilization. To enable barge and truck operators to plan in cooperation, we propose the novel method *Departure Learning* (DL). With DL a barge operator learns from feedback from a truck operator what barge schedule will lead to good performance of their shared synchromodal transport system. The simulation results show that DL performs better than the fixed schedule case. DL uses random variables to choose the set of schedules the truck operator should evaluate, causing the solution quality to vary. Future research should analyse this sensitivity and the influence of the learning parameters.

The shown framework assumes only one barge and one truck company. It can be extended to several truck companies by using the weighted sum of their reported performances. Enabling the barge company to operate more barges and serve more than one terminal is important future research for making the proposed framework fit more complex networks. Furthermore, in reality, barge operators often have direct contacts to shippers and logistics providers. Future research could extend the framework to the case where the cooperation between the barge and truck operator only consider time-varying excess capacity. DL enables cooperative planning and relies on the honesty of the cooperating parties. Future research into sharing mechanisms and control schemes will make the framework more robust towards abuse.

References

1. Behdani, B., Fan, Y., Wiegmans, B., Zuidwijk, R.: Multimodal schedule design for synchromodal freight transport systems. Eur. J. Transp. Infrastruct. Res. **16**(3), 424–444 (2016)
2. Cruijssen, F., Dullaert, W., Fleuren, H.: Horizontal cooperation in transport and logistics: a literature review. Transp. J. **46**, 22–39 (2007)
3. Demir, E., Burgholzer, W., Hrušovskỳ, M., Arikan, E., Jammernegg, W., Van Woensel, T.: A green intermodal service network design problem with travel time uncertainty. Transp. Res. Part B: Methodol. **93**, 789–807 (2016)
4. van Essen, H., et al.: Handbook on the external costs of transport (2019). www.cedelft.eu
5. Eurostat: Database: annual road freight transport vehicle movements, loaded and empty, by reporting country
6. Larsen, R.B., Atasoy, B., Negenborn, R.R.: Simultaneous planning of container and vehicle-routes using model predictive control. In: Proceedings of European Control Conference, pp. 2177–2182 (2019)
7. Larsen, R.B., Atasoy, B., Negenborn, R.R.: Model predictive control for simultaneous planning of container and vehicle routes. Eur. J. Control (2020)
8. Li, L., Negenborn, R.R., De Schutter, B.: Distributed model predictive control for cooperative synchromodal freight transport. Transp. Res. Part E: Logist. Transp. Rev. **105**, 240–260 (2017)

9. Li, L., Negenborn, R.R., Schutter, B.D.: Intermodal freight transport planning - a receding horizon control approach. Transp. Res. Part C: Emerg. Technol. **60**, 77–95 (2015)
10. Löfberg, J.: YALMIP: a toolbox for modeling and optimization in Matlab. In: Proceedings of International Symposium on Computer Aided Control System Design (2004)
11. Nabais, J.L., Negenborn, R.R., Botto, M.A.: 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 (2013)
12. Psaraftis, H.N., Wen, M., Kontovas, C.A.: Dynamic vehicle routing problems: three decades and counting. Networks **67**(1), 3–31 (2016)
13. Qu, W., Rezaei, J., Maknoon, Y., Tavasszy, L.: Hinterland freight transportation replanning model under the framework of synchromodality. Transp. Res. Part E: Logist. Transp. Rev. **131**, 308–328 (2019)
14. Sager, S., Bock, H.G., Diehl, M.: The integer approximation error in mixed-integer optimal control. Math. Program. **133**(1), 1–23 (2012)
15. SteadieSeifi, M., Dellaert, N.P., Nuijten, W., Van Woensel, T., Raoufi, R.: Multimodal freight transportation planning: a literature review. Eur. J. Oper. Res. **233**(1), 1–15 (2014)
16. Van Riessen, B., Negenborn, R.R., Dekker, R., Lodewijks, G.: Service network design for an intermodal container network with flexible transit times and the possibility of using subcontracted transport. Int. J. Shipp. Transp. Logist. **7**(4), 457–478 (2015)
17. Xu, S.X., Huang, G.Q., Cheng, M.: Truthful, budget-balanced bundle double auctions for carrier collaboration. Transp. Sci. **51**(4), 1365–1386 (2017)
18. Xu, Y., Cao, C., Jia, B., Zang, G.: Model and algorithm for container allocation problem with random freight demands in synchromodal transportation. Math. Probl. Eng. **2015**, 1–13 (2015)