A dynamic shipment matching problem in hinterland synchromodal transportation

Guo, Wenjing; Atasoy, Bilge; van Blokland, Wouter Beelaerts; Negenborn, Rudy R.

DOI
10.1016/j.dss.2020.113289

Publication date
2020

Document Version
Final published version

Published in
Decision Support Systems

Decision Support Systems 134 (2020) 113289

Contents lists available at ScienceDirect

Decision Support Systems

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

A dynamic shipment matching problem in hinterland synchromodal transportation

Wenjing Guo*, Bilge Atasoy, Wouter Beelaerts van Blokland, Rudy R. Negenborn

Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, 2628CD Delft, the Netherlands

ARTICLE INFO

Keywords:
Hinterland synchromodal transportation
Dynamic shipment matching
Rolling horizon approach
Heuristic algorithm

ABSTRACT

Hinterland intermodal transportation is the movement of containers between deep-sea ports and inland terminals by using trucks, trains, barges, or any combination of them. Synchromodal transportation, as an extension of intermodal transportation, refers to transport systems with dynamic updating of plans by incorporating real-time information. The trend towards spot markets and digitalization in hinterland intermodal transportation gives rise to online synchromodal transportation problems. This paper investigates a dynamic shipment matching problem in which a centralized platform provides online matches between shipment requests and transport services. We propose a rolling horizon approach to handle newly arrived shipment requests and develop a heuristic algorithm to generate timely solutions at each decision epoch. The experiment results demonstrate the solution accuracy and computational efficiency of the heuristic algorithm in comparison to an exact algorithm. The proposed rolling horizon approach outperforms a greedy approach from practice in total costs under various scenarios of the system.

1. Introduction

Hinterland intermodal transportation is the movement of containers between deep-sea ports and inland terminals by using trucks, trains, barges, or any combination of them [17]. Compared with unimodal transportation, intermodal transportation has the flexibility to use different modes considering the specific characteristics of containers and in turn achieves better performance in costs, delays, and emissions [6]. However, due to the utilization of multiple modes, operating an intermodal transportation system is very complex. In intermodal transportation, barge and train services normally follow fixed time schedules and have limited free capacity [6]. Conversely, truck services are usually not scheduled and have time-dependent travel times as a result of road traffic congestion [18]. Therefore, constraints such as time compatibility between different services and capacity limitations of barge and train services need to be considered in intermodal transport planning.

Synchromodal transportation, as an extension of intermodal transportation, refers to transport systems with dynamic updating of planning by incorporating real-time information [9]. The trend towards spot markets and digitalization in hinterland intermodal transportation increases the need for such online synchromodal transportation problems. In the literature, most of the existing studies assume that container shipments are only collected from large shippers based on long-term contracts. These contractual shipment requests are often fixed and known over a given planning period. Recently, quite a few studies [e.g.,21,22] have pointed out the trend towards spot markets in container transportation. Different from the former contracted requests, spot shipment requests arrive in real-time and require receiving transport solutions as soon as possible. Thanks to the development of digitalization and advanced information and communication technologies in logistic industries, information can be collected in real-time, and decisions can be made online [14]. Nevertheless, these new trends also introduce complexity in intermodal transport planning, unveiling the need for decision support systems adapted to dynamic contexts.

In this paper, we investigate a dynamic shipment matching (DSM) problem in which a platform provides online matches between shipment requests and transport services. We consider an online synchromodal matching platform that receives contractual and spot shipment requests from shippers, and receives transport services from carriers, as shown in Fig. 1. Shippers are the entities that are searching for services to transport their shipments. Examples of shippers include freight forwarders and ocean carriers. Carriers are the entities that provide transport services. Carriers could be truck, train or barge companies. We consider a network operator as the owner of the platform. A network operator could be a logistics service provider or an alliance

* Corresponding author.
E-mail addresses: W.Guo-2@tudelft.nl (W. Guo), B.Atasoy@tudelft.nl (B. Atasoy), W.W.A.BeelaertsvanBlokland@tudelft.nl (W.B. van Blokland), R.R.Negenborn@tudelft.nl (R.R. Negenborn).

https://doi.org/10.1016/j.dss.2020.113289
Received 3 September 2019; Received in revised form 19 March 2020; Accepted 22 March 2020
Available online 15 May 2020

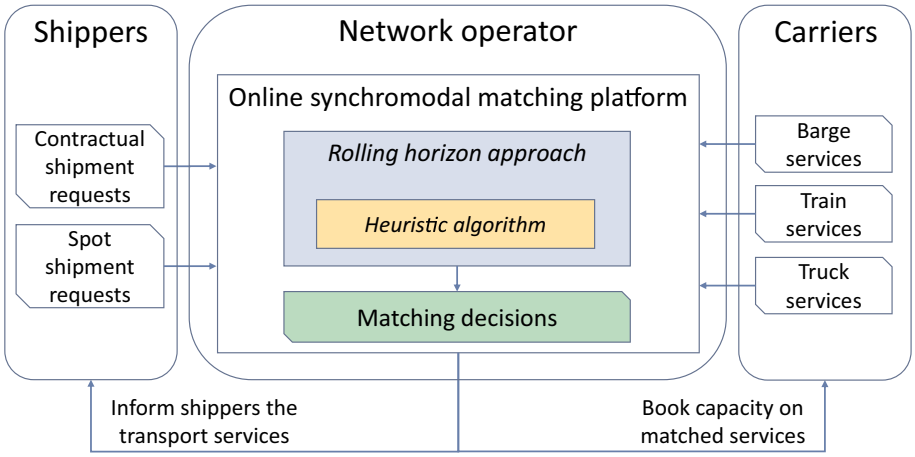

Fig. 1. Illustration of an online synchromodal matching platform. The platform provides online matches between shipment requests received from shippers and transport services received from carriers thanks to the developed rolling horizon approach.

formed by multiple carriers. The recent developments in information technologies such as cloud computing and Internet of Things allow real-time information sharing and container tracking, which facilitates the adoption of such a platform in practice.

The objective of the platform is to minimize the total cost of matching shipment requests and transport services over a given planning horizon. Due to the capacity limitation of barge and train services, decisions made for current requests may influence the decisions for future requests. Therefore, dynamic approaches that create online matching decisions for current requests are required. In this paper, we design a rolling horizon approach to handle dynamically revealed shipment requests and develop a heuristic algorithm to solve the DSM problem in a computationally efficient way.

The remainder of this paper is structured as follows. We discuss the relevant literature in Section 2. In Section 3, we formally describe the DSM problem. In Section 4, we explain the implementation of dynamic approaches. In Section 5, we present optimization algorithms. In Section 6, we describe the generation of instances and present the experiment results. Finally, in Section 7, we provide concluding remarks and directions for future research.

2. Literature review

Over the past decades, different freight transport concepts have been proposed in the literature and in the industry: multimodality, intermodality, co-modality, and synchromodality [9]. Although these concepts are often used interchangeably, there are subtle differences between these terms: multimodality focuses on the utilization of multiple modes; intermodality emphasizes the integration between different modes by using standard loading units; co-modality aims to have efficient utilization of resources; synchromodality, as an extension of intermodality, adds dynamic updating of transport plans over a network to benefit from real-time information [1]. In this section, the studies related to the DSM problem have been divided into two categories: hinterland intermodal transportation and synchromodality.

2.1. Hinterland intermodal transportation

Hinterland intermodal transportation is the provision of efficient, reliable, and sustainable services through integrated strategic and tactical planning at a network level. Strategic planning concerns the design of transportation network topologies, such as direct link, corridor, or hub-and-spoke [5]. Konings et al. [10] investigate the benefits of a hub-and-spoke network for hinterland transportation in turnaround times, waiting times, and the reliability of barge services. Containers at a seaport terminal that have different destinations in the hinterland would be transported together to the hub and after being regrouped and bundled with containers originate from other seaport terminals would continue their trip to their inland destination.

Tactical planning refers to optimally utilizing the given network by choosing transportation services, allocating their capacity to customer demands, and planning their itineraries and frequency [17]. Bhattacharya et al. [3] propose a mixed integer programming model to optimize schedules for an intermodal transport network by taking into account the road traffic flow estimation. Zuidwijk and Veenstra [23] propose a single period model to allocate containers to a truck or barge and schedule the barge departure time considering container release time uncertainty and service transit time uncertainty. Crainic et al. [4] propose a service network design model to decide the optimal schedules for the services operated by a fleet of shuttles on the railway network connecting seaport terminals and inland terminals. Demir et al. [6] investigate a service network design problem with travel time uncertainty to decide on the routing of containers and the departure time of transport services.

2.2. Synchromodality

While intermodality focuses on offline planning in which all forms of input information are required in advance and decisions are made before the start of transportation, synchromodality emphasizes online planning in which real-time information about the current state of the transport system can be taken into account in online planning processes [7]. Specifically, synchromodal transport planning deals with dynamic events that are not explicitly addressed in intermodal transportation, including the representation of real-time data, decisions, and system states [5]. The most common dynamic events are the arrival of new shipment requests, but container flows and travel times are possible dynamics as well.

In the literature, Fazi et al. [8] develop a decision support system for the optimal allocation of import containers to a heterogeneous fleet composed of barges and trucks. van Riessen et al. [19] design a decision tree to derive real-time decision rules for suitable allocation of containers to services. Rivera and Mes [16] propose an algorithm based on approximate dynamicm programming to assign newly arrived containers to either a barge or a truck. Although the above studies considered the utilization of multiple modes, none of them take into account the transshipment operations between different services. Research that models transshipment in synchromodal transportation, such as Li et al. [11] and Qu et al. [15], are usually designed for container flows. However, in practice, shippers would like to receive their shipments as a whole. Therefore, in this paper, we investigate the DSM problem from shipment requests' perspective, namely, decisions are

designed as binary variables indicating the allocation of a specific shipment request to a specific service. Mes and Iacob [12] propose a greedy approach to select the cheapest services for dynamically arrived shipment requests but without the consideration of road traffic congestions. Due to the limited capacity of road infrastructures, traffic congestions exist during several periods of a day [18]. The variation of road travel times has been well investigated in the literature and therefore can be incorporated in the online synchromodal matching process.

2.3. Contributions

In the literature, the work most similar to our work is Li et al. [11], which proposes a rolling horizon approach to control container flows in a hinterland intermodal network by considering time-dependent truck travel times and time-schedules for trains and barges. In contrast to our work, Li et al. [11] focuses on aggregated container flows instead of specific shipment requests with time windows, and therefore uses the value of time instead of delay costs in the objective function to push containers move to their destinations.

The main contributions of this paper are as follows. First, we propose a rolling horizon approach to handle newly arrived shipment requests. The implementation of the rolling horizon approach relies on an optimization algorithm that can generate timely matching decisions at each decision epoch. In particular, we develop a heuristic algorithm to solve the DSM problem. Third, we conduct extensive experiments to assess the performance of the heuristic algorithm in comparison to an exact algorithm, and the performance of the rolling horizon approach in comparison to a greedy approach from practice. Briefly, we design, operationalize and validate an online matching platform in the context of synchromodal transportation.

3. Problem description

Let R be the set of shipment requests. Each shipment request $r \in R$ is characterized by its announce time $\Gamma_r^{\text{announce}}$ (i.e., the time when the platform receives the request), release time $\Gamma_r^{\text{release}}$ (i.e., the time when the shipment is available for hinterland transportation) at origin terminal o_r, due time Γ_r^{due} (i.e., the time that the shipment needs to be delivered) at destination terminal d_r, and container volume q_r (i.e., the number of containers). Delay in delivery is available but with a delay cost coefficient per container per hour overdue c_r^{delay}. The lead time of shipment request r is represented as, $LD_r = \Gamma_r^{\text{due}} - \Gamma_r^{\text{release}}$.

Shipment requests can be divided into two groups: contractual requests R^{contract} and spot requests R^{spot}. For a contractual request $r \in R^{\text{contract}}$, the network operator has long-term contracts with shippers. Therefore, the announce time of contractual request r is, $\Gamma_r^{\text{announce}} = 0$. All the information $\{o_r, d_r, q_r, \Gamma_r^{\text{release}}, \Gamma_r^{\text{due}}, c_r^{\text{delay}}\}$ is known in a given planning horizon. Conversely, for a spot request $r \in R^{\text{spot}}$, the platform receives the request from spot markets in real-time. The information of the spot request $\{o_r, d_r, q_r, \Gamma_r^{\text{release}}, \Gamma_r^{\text{due}}, c_r^{\text{delay}}\}$ is unknown before its announce time.

Let S be the set of transportation services. According to the type of modes, services can be divided into two groups: time-scheduled barge and train services, and departure time flexible truck services.

Barge and train services have limited capacity and fixed time schedules but can help generating economies of scale. Each barge or train service $s \in S^{\text{barge}} \cup S^{\text{train}}$ is characterized by its origin terminal o_s, destination terminal d_s, free capacity in terms of loading units (i.e., containers) Q_s, departure time (at origin terminal) TD_s, arrival time (at destination terminal) TA_s, transport cost c_s, and generation of carbon emissions e_s.

Truck services have unlimited capacity, flexible departure times, and time-dependent travel times $t_s^{\text{truck}}(\gamma) = \theta_s^m \gamma + \eta_s^m$, $\forall\, \gamma \in T^m$, as shown in Fig. 2. Here, we let γ be the departure time of truck services, and $\mathbb{T}$ represents the set of time periods within a day. A time period can

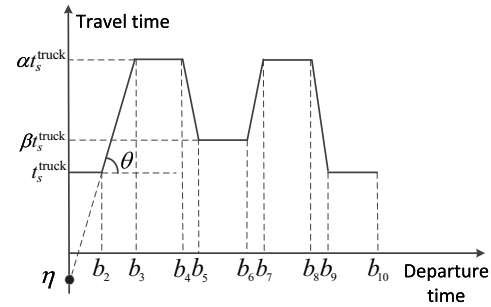

Fig. 2. Time-dependent travel times of truck services.

be defined by two consecutive breakpoints. Let t_s^{truck} be the travel time at non-peak periods, α and β be traffic congestion coefficients. For time period $T^2 = [b_2, b_3]$, given the values b_2, b_3, t_s^{truck}, $\alpha t_s^{\text{truck}}$, we can calculate the slope θ of the function and the intersection η with the y-axis. Each truck service $s \in S^{\text{truck}}$ is characterized by its origin o_s, destination d_s, time-dependent travel time $t_s^{\text{truck}}(\gamma)$, transport cost c_s, and generation of carbon emissions e_s.

As spot shipment requests arrive in real-time, the platform provides online matches between shipment requests and transport services. A match is defined as a combination of a shipment and a service, which means the shipment will be transported by the service from the service's origin to the service's destination. Each shipment might be matched with multiple services, each service might be matched with multiple shipments. An illustrative example of shipment matching in synchromodal transportation is shown in Fig. 3. Matching decision $\langle r1, s4 \rangle$ means shipment $r1$ will be transported by service $s4$ from terminal 1 to terminal 5; matching decision $\langle r2, s1 \rangle$, $\langle r2, s3 \rangle$, $\langle r2, s7 \rangle$ means shipment $r2$ will be transported by service combination $[s1, s3, s7]$ from terminal 1 to terminal 6.

To model this problem, we make the following five assumptions. First, we assume the platform is centralized and the contracts among carriers, shippers, terminal operators, and the network operator have been made. Therefore, we do not consider fairness, pricing, and contracting strategies among players. Second, we do not model the accept/reject decisions and consider only the accepted spot requests by the platform. Third, we assume that shippers require their shipments to be transported as a whole, thus shipments are unsplittable. Fourth, we assume shippers require to receive matching decisions before the release time of shipments. Therefore, the response time of request r is $\Delta\Gamma_r = \Gamma_r^{\text{release}} - \Gamma_r^{\text{announce}}$. Fifth, we assume the capacity of truck services is unlimited. Therefore, the synchromodal matching system always has feasible matches for newly arrived shipment requests. Last, we do not consider stochasticity of travel times in this paper. Instead, we use deterministic travel times for all services, and consider time-dependent travel times for trucks, since the road traffic patterns have

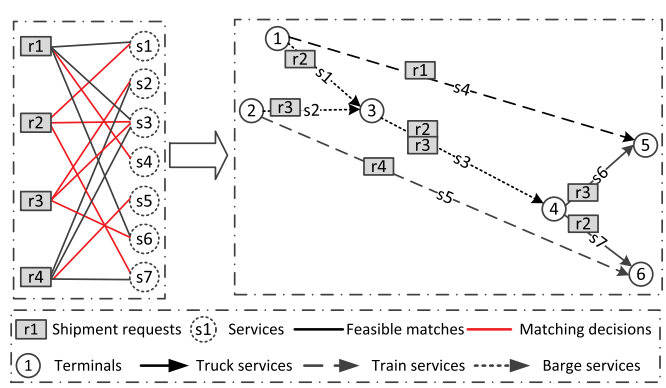

Fig. 3. Illustrative example of shipment matching in synchromodal transportation.

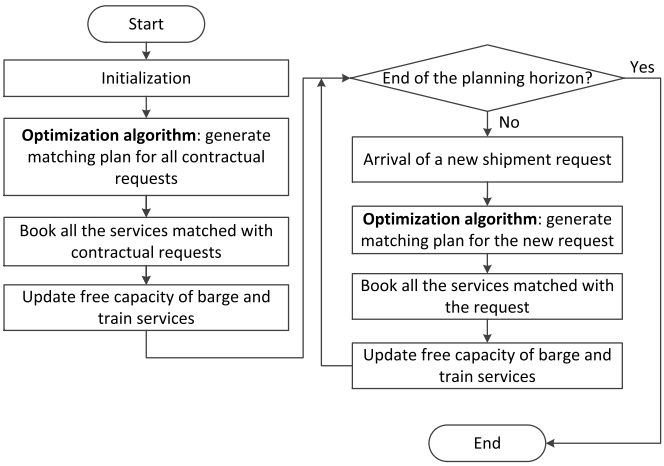

Fig. 4. Flow chart of the greedy approach.

been well investigated in the literature [18].

4. Dynamic approaches

To handle newly arrived shipment requests, we need to design methodologies that can update the decisions based on dynamically revealed information. This paper proposes a rolling horizon approach for the DSM problem and uses a greedy approach as the benchmark. While the greedy approach makes matching decisions for each newly arrived shipment request and the decisions are fixed once they are made, the rolling horizon approach makes decisions at fixed time points for all active requests including newly received requests at the current time interval and the requests received at previous time intervals which have not expired yet, and the decisions are fixed only when the response for the request cannot be further postponed, namely, the request will expire before the next decision epoch.

4.1. Benchmark: greedy approach

Greedy approach (GA) is a simple, intuitive algorithm that makes fixed decisions at each step. In practice, a GA is often used for container transport planning [19]. By using the GA, a shipment request is assigned to the cheapest feasible service at the time of request arrival. Fig. 4 presents the flow chart of the GA applied in dynamic shipment matching. Specifically, the platform provides matches for all the contractual requests received before the planning horizon. After that, the platform books all the services matched with the contractual requests and updates the free capacity of barges and trains. A dynamic event, that is the arrival of a spot shipment request before the end of the planning horizon, triggers a new optimization process. After that, the platform books all the services matched with the spot shipment request, and updates the free capacity of barges and trains.

4.2. Rolling horizon approach

Rolling horizon approach (RHA) is a periodic reoptimization approach, which has been applied in many research fields, such as ride-sharing problems [13] and parcel delivery problems [2]. Under a RHA, the system is optimized periodically at pre-specified points in time called *optimization times*. The length between two consecutive optimization times is called *the optimization interval*, h. The RHA is therefore executed at a given set of time points $\{0, h, 2h, \ldots, T\}$. Here, T is the length of the planning horizon.

Under the RHA, plans are made using all known information within a planning horizon, but decisions are not finalized until necessitated by a deadline. Re-optimizing the system allows for enhancing the reliability of the system and improving its performance by incorporating the latest information. The flow chart of the RHA applied in the DSM problem is presented in Fig. 5. At each decision epoch, the system determines the matches for all active shipments. At time point t, shipment r is active if its announce time is earlier than t, and its release time is later than t. The matching plan for active shipment r made at time point t is fixed only if its release time is earlier than $t + h$, namely, the shipment request will expire before the next decision epoch. Thus, the system books all the services matched with this request, and updates the free capacity of barge and train services.

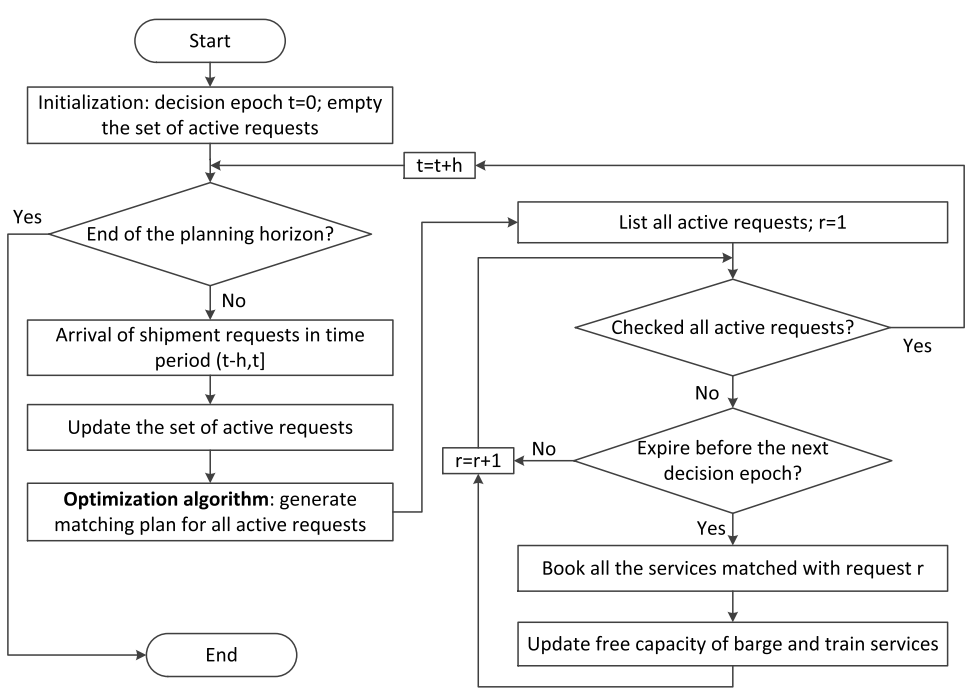

Fig. 5. Flow chart of the rolling horizon approach.

Table 1
Notations used in this paper.

Sets	
N	Terminals
R	Shipment requests
S	Transport services, $S = S^{\text{barge}} \cup S^{\text{train}} \cup S^{\text{truck}}$
S_i^+	Transport services depart at terminal $i \in N$, $S_i^+ = S_{i^+}^{\text{barge}} \cup S_{i^+}^{\text{train}} \cup S_{i^+}^{\text{truck}}$
S_i^-	Transport services arrive at terminal $i \in N$, $S_i^- = S_{i^-}^{\text{barge}} \cup S_{i^-}^{\text{train}} \cup S_{i^-}^{\text{truck}}$
Parameters	
o_r	Origin terminal of shipment request $r \in R$
d_r	Destination terminal of shipment request $r \in R$
q_r	Container volume of shipment request $r \in R$
$\Gamma_r^{\text{announce}}$	Announce time of shipment request $r \in R$
$\Gamma_r^{\text{release}}$	Release time of shipment request $r \in R$
Γ_r^{due}	Due time of shipment request $r \in R$
c_r^{delay}	Delay cost coefficient of request $r \in R$ per container per hour overdue
o_s	Origin terminal of service $s \in S$
d_s	Destination terminal of service $s \in S$
Q_s	Free capacity of service $s \in S^{\text{barge}} \cup S^{\text{train}}$
TD_s	Departure time of service $s \in S^{\text{barge}} \cup S^{\text{train}}$
TA_s	Arrival time of service $s \in S^{\text{barge}} \cup S^{\text{train}}$
t_s^{truck}	Travel time of truck service $s \in S^{\text{truck}}$ at non-peak periods
α, β	Road traffic congestion coefficients
b_k	The kth breakpoint of time-dependent travel time functions of truck services, $k = \{1,2,\ldots,K\}$
T^m	The mth time period within a day, $T^m = [b_m, b_{m+1}]$, $m = \{1,2,\ldots,K-1\}$
θ_s^m	The slope of the travel time function of truck service s for time period T^m
η_s^m	The intersection of the travel time function of truck service $s \in S^{\text{truck}}$ for time period T^m
e_s	Carbon emissions of service $s \in S$ per container
c_s	Transport cost of service $s \in S$ per container
lc^{barge}	Loading/unloading cost of barge services
lt^{barge}	Loading/unloading time of barge services
lc^{train}	Loading/unloading cost of train services
lt^{train}	Loading/unloading time of train services
lc^{truck}	Loading/unloading cost of truck services
lt^{truck}	Loading/unloading time of truck services
c^{storage}	Storage cost coefficient at terminals per container per hour
c^{emission}	carbon tax coefficient per ton
M	Large (enough) numbers used for binary constraints
Variables	
x_{rs}	A binary variable equal to 1 if request $r \in R$ is matched with service $s \in S$, 0 otherwise
A_{ri}	Arrival time of request $r \in R$ at terminal $i \in N$
f_{ri}^+	Loading cost of request $r \in R$ at terminal $i \in N$ per container
f_{ri}^-	Unloading cost of request $r \in R$ at terminal $i \in N$ per container
w_{ri}	Storage time of request $r \in R$ at terminal $i \in N$
Γ_r^{delay}	Delay of request $r \in R$ at destination terminal d_r
t'_{rs}	Travel time of truck service $s \in S^{\text{truck}}$ with request $r \in R$
τ_{rs}	Departure time of truck service $s \in S^{\text{truck}}$ with request $r \in R$
τ'_{rs}	Normalized departure time of truck service $s \in S^{\text{truck}}$ with request $r \in R$, $0 \le \tau'_{rs} \le 24$
n_{rs}	An integer variable used for normalizing departure time of truck service $s \in S^{\text{truck}}$ with request $r \in R$
ζ_{rs}^k	A continuous variable used for linearizing the time-dependent travel time function of truck service $s \in S^{\text{truck}}$, $0 \le \zeta_{rs}^k \le 1$
ξ_{rs}^m	A binary variable used for linearizing the time-dependent travel time function of truck service $s \in S^{\text{truck}}$

5. Optimization algorithms

In this section, we present two optimization algorithms to solve the DSM problem: an exact algorithm and a heuristic algorithm. While the exact algorithm aims to generate optimal solutions, the heuristic algorithm is designed to generate timely solutions. The notations used in this paper are shown in Table 1.

5.1. Exact algorithm

In this section, we present a mixed integer linear programming model (MILP) for the DSM problem. The MILP model is solved by an exact algorithm which is the CPLEX solver. The objective function (Eq. (1)) minimizes the total costs for the matching of all shipments with services. The total costs consist of transport costs (including transit costs, transfer costs, and storage costs), delay costs, and carbon tax. We include delay costs to address the level of services (i.e., delayed deliveries). Considering carbon tax follows the trend towards sustainability in the transport industry. In the literature, there exist several models for calculating emission charges. However, most of the models require detailed input data (e.g., the mass of the vehicle, air, and rolling resistance) which is in many cases not available. As an alternative, the activity-based method that multiplies the number of containers with the CO_2 emission factor yields better feasibility in transportation practice and has been applied in many studies [6,18]. Therefore, this paper uses the activity-based method to charge CO_2 emissions.

Minimize

$$\sum_{r\in R}\sum_{s\in S} x_{rs} q_r c_s + \sum_{r\in R}\sum_{i\in N} (f_{ri}^+ + f_{ri}^-) q_r + \sum_{r\in R}\sum_{i\in N} w_{ri} q_r c^{\text{storage}} + \sum_{r\in R} \Gamma_r^{\text{delay}} q_r c_r^{\text{delay}} + \sum_{r\in R}\sum_{s\in S} x_{rs} e_s q_r c^{\text{emission}} \tag{1}$$

subject to

$$\sum_{s\in S_{o_r}^+} x_{rs} = 1, \quad \forall\, r \in R, \tag{2}$$

$$\sum_{s\in S_{d_r}^-} x_{rs} = 1, \quad \forall\, r \in R, \tag{3}$$

$$\sum_{s\in S_i^+} x_{rs} = \sum_{s\in S_i^-} x_{rs}, \quad \forall\, r \in R,\ i \in N\backslash\{o_r, d_r\}, \tag{4}$$

$$\sum_{r\in R} x_{rs} q_r \le Q_s, \quad \forall\, s \in S^{\text{barge}} \cup S^{\text{train}}, \tag{5}$$

$$f_{ri}^+ = \sum_{s\in S_{i^+}^{\text{barge}}} x_{rs} lc^{\text{barge}} + \sum_{s\in S_{i^+}^{\text{train}}} x_{rs} lc^{\text{train}} + \sum_{s\in S_{i^+}^{\text{truck}}} x_{rs} lc^{\text{truck}}, \quad \forall\, r \in R,\ i \in N\backslash\{d_r\}, \tag{6}$$

$$f_{ri}^- = \sum_{s\in S_{i^-}^{\text{barge}}} x_{rs} lc^{\text{barge}} + \sum_{s\in S_{i^-}^{\text{train}}} x_{rs} lc^{\text{train}} + \sum_{s\in S_{i^-}^{\text{truck}}} x_{rs} lc^{\text{truck}}, \quad \forall\, r \in R,\ i \in N\backslash\{o_r\}, \tag{7}$$

$$A_{ro_r} = \Gamma_r^{\text{release}}, \quad \forall\, r \in R, \tag{8}$$

$$A_{ri} \le (TA_s + lt^{\text{barge}}) x_{rs} + M(1 - x_{rs}), \quad \forall\, r \in R,\ i \in N\backslash\{o_r\},\ s \in S_{i^-}^{\text{barge}}, \tag{9}$$

$$A_{ri} \ge (TA_s + lt^{\text{barge}}) x_{rs} + M(x_{rs} - 1), \quad \forall\, r \in R,\ i \in N\backslash\{o_r\},\ s \in S_{i^-}^{\text{barge}}, \tag{10}$$

$$A_{ri} \le (TA_s + lt^{\text{train}}) x_{rs} + M(1 - x_{rs}), \quad \forall\, r \in R,\ i \in N\backslash\{o_r\},\ s \in S_{i^-}^{\text{train}}, \tag{11}$$

$$A_{ri} \ge (TA_s + lt^{\text{train}}) x_{rs} + M(x_{rs} - 1), \quad \forall\, r \in R,\ i \in N\backslash\{o_r\},\ s \in S_{i^-}^{\text{train}}, \tag{12}$$

$$A_{ri} \le \tau_{rs} + t'_{rs} + lt^{\text{truck}} x_{rs} + M(1 - x_{rs}),$$
$$\forall\, r \in R,\ i \in N\backslash\{o_r\},\ s \in S_{i^-}^{\text{truck}}, \tag{13}$$

$$A_{ri} \ge \tau_{rs} + t'_{rs} + lt^{\text{truck}} x_{rs} + 2M(x_{rs} - 1),$$
$$\forall\, r \in R,\ i \in N\backslash\{o_r\},\ s \in S_{i^-}^{\text{truck}}, \tag{14}$$

$$A_{ri} \le (TD_s - lt^{\text{barge}}) x_{rs} + M(1 - x_{rs}),$$
$$\forall\, r \in R,\ i \in N\backslash\{d_r\},\ s \in S_{i^+}^{\text{barge}}, \tag{15}$$

$$A_{ri} \le (TD_s - lt^{\text{train}}) x_{rs} + M(1 - x_{rs}),$$
$$\forall\, r \in R,\ i \in N\backslash\{d_r\},\ s \in S_{i^+}^{\text{train}}, \tag{16}$$

$$A_{ri} \le \tau_{rs} - lt^{\text{truck}} x_{rs} + M(1 - x_{rs}),$$
$$\forall\, r \in R,\ i \in N\backslash\{d_r\},\ s \in S_{i^+}^{\text{truck}}, \tag{17}$$

$$\tau'_{rs} = \tau_{rs} - 24 n_{rs}, \quad \forall\, r \in R,\ s \in S^{\text{truck}}, \tag{18}$$

$$\tau'_{rs} = \sum_k \zeta_{rs}^k b_k, \quad \forall\, r \in R,\ s \in S^{\text{truck}}, \tag{19}$$

$$\sum_k \zeta_{rs}^k = 1, \quad \forall\, r \in R,\ s \in S^{\text{truck}}, \tag{20}$$

$$t'_{rs} = \sum_k \zeta_{rs}^k (\theta_s^m b_k + \eta_s^m), \quad \forall\, r \in R,\ s \in S^{\text{truck}}, \tag{21}$$

$$\sum_m \xi_{rs}^m = 1, \quad \forall\, r \in R,\ s \in S^{\text{truck}}, \tag{22}$$

$$\zeta_{rs}^1 \le \xi_{rs}^1, \quad \forall\, r \in R,\ s \in S^{\text{truck}}, \tag{23}$$

$$\zeta_{rs}^k \le \xi_{rs}^{k-1} + \xi_{rs}^k,$$
$$\forall\, r \in R,\ s \in S^{\text{truck}},\ k \in \{2, 3, \ldots K - 1\}, \tag{24}$$

$$\zeta_{rs}^K \le \xi_{rs}^{K-1}, \quad \forall\, r \in R,\ s \in S^{\text{truck}}, \tag{25}$$

$$w_{ri} \ge (TD_s - lt^{\text{barge}}) x_{rs} - A_{ri},$$
$$\forall\, r \in R,\ i \in N\backslash\{d_r\},\ s \in S_{i^+}^{\text{barge}}, \tag{26}$$

$$w_{ri} \ge (TD_s - lt^{\text{train}}) x_{rs} - A_{ri},$$
$$\forall\, r \in R,\ i \in N\backslash\{d_r\},\ s \in S_{i^+}^{\text{train}}, \tag{27}$$

$$w_{ri} \ge \tau_{rs} - lt^{\text{truck}} x_{rs} + M(x_{rs} - 1) - A_{ri},$$
$$\forall\, r \in R,\ i \in N\backslash\{d_r\},\ s \in S_{i^+}^{\text{truck}}, \tag{28}$$

$$\Gamma_r^{\text{delay}} \ge A_{rd_r} - \Gamma_r^{\text{due}}, \quad \forall\, r \in R. \tag{29}$$

Constraints (2)–(4) manage the inflow of shipments at their origin terminal, outflow at destination terminal, and flow conservation at transshipment terminal. Constraints (5) ensure that the total container volumes of shipments carried by service $s \in S^{\text{barge}} \cup S^{\text{train}}$ do not exceed its free capacity. Constraints (6)–(7) represent the loading and unloading cost of request r per container generated at terminal i. Constraints (8) assume that the arrival time of request r at origin terminal is the release time. Constraints (9)–(12) ensure that the arrival time of request r at terminal i is the arrival time of service $s \in S^{\text{barge}} \cup S^{\text{train}}$ plus unloading time, if request r is transported by service s entering terminal i. Constraints (13)–(14) ensure that the arrival time of request r at terminal i is the sum of departure time of service $s \in S^{\text{truck}}$ with request r at terminal o_s, travel time of truck service s, and unloading time, if request r is transported by truck service s entering terminal i. In constraints (14), we use $2M$ instead of M in the right-hand side to make sure the value of A_{ri} will not be influenced by the constraints when $x_{rs} = 0$ and $\tau_{rs} = M$. Constraints (15)–(17) ensure that the arrival time of request r at terminal i is earlier than the departure time of service $s \in S$ minus loading time, if request r is transported by service s leaving terminal i. Constraints (18)–(25) are imposed to linearize the time-dependent travel time functions of truck services. Constraints (26)–(28) ensure that the storage time of request r at terminal i is the departure time of service s minus the arrival time of request r at terminal i and minus loading time, if request r is transported by service s leaving terminal i. Constraints (29) are imposed to calculate the late deliveries of request r at destination terminal d_r. We do not penalize earlier deliveries but only late deliveries.

5.2. Heuristic algorithm

Due to the computational complexity of the matching problem, the exact algorithm proposed in Section 5.1 approach cannot generate feasible solutions for realistic instances. Therefore, this paper proposes a preprocessing-based heuristic algorithm to reduce the computational complexity. The algorithm consists of three steps: preprocessing of path generation in which no request-specific characteristics are taken into account, preprocessing of feasible matches in which request-specific characteristics (i.e., release time and due time) are considered, and binary integer programming to generate 'optimal' solutions.

5.2.1. Preprocessing of path generation

We define a path as a combination of services. A path p can consist of a single service or multiple services. For example, a path p consists of a barge service s_1 and a truck service s_2, thus, $p = [s_1, s_2]$. We define L as the largest number of services in a path. Due to fixed schedules of barge and train services, some of the service combinations are infeasible. Let $P_{ij}^{\ l}$ be the set of feasible paths with l services that depart at terminal $i \in N$ and arrive at terminal $j \in N$, $l \in \{1,\ldots,L\}$. A path $p \in P_{ij}^{\ l}$ is feasible only if all the services in path $p = [s_1, \ldots, s_l]$ satisfies spatial and time compatibility: for service s_n, $s_{n+1} \in p$, $n \in \{1,\ldots,l - 1\}$, the destination terminal of service s_n should be the same as the origin terminal of service s_{n+1}; the arrival time of s_n plus unloading and loading time at the transshipment terminal should be earlier than the departure time of service s_{n+1}.

Based on the above principles, feasible paths with maximum L services are generated by using the offline preprocessing algorithm presented in Algorithm 1. The algorithm starts with determining the feasible paths for each origin-destination pair with just one service, and subsequently combines these paths with a single service to create feasible paths with two services, three services, and so on. For each feasible path, we record the virtual departure and arrival time points of all the services in the path by calling the AUXILIARYTIMEPOINTS as described in Algorithm 2. The virtual departure (arrival) time points of barge and train services are the departure (arrival) time of these services minus (plus) loading (unloading) time. Instead of determining the departure time of truck services to avoid traffic congestion, we define the virtual departure time points of truck services as the virtual arrival time points of their previous services to reduce computational complexity. The time-dependent travel time of truck services is calculated based on the virtual departure time point plus loading time. To examine whether a path $p' = [s_1, \ldots, s_{l-1}, s] \in P_{ij}^{\ l}$ is feasible, we check the time compatibility between path $p = [s_1, \ldots, s_{l-1}] \in P_{io_s}^{\ l-1}$ and service $s \in S_j^-$ by calling the TIMECOMPATIBLE1 as described in Algorithm 3.

Algorithm 1. Path generation algorithm.

Input: Set of transportation services S, set of terminals N, the largest number of services in a path L, index $l \in \{1, 2, ..., L\}$.
Output: Set of feasible paths $P = P^1 \cup ... P^l \cup ... P^L$, $P^l_{ij} \subseteq P^l$ represents the set of feasible paths with l services that depart at node i, and arrive at node j. Auxiliary time points $MT^l_p = \left[MT^{l1}_p, ..., MT^{l(2l)}_p\right]$.
Initialize: Let $P \leftarrow \emptyset, MT^l_p \leftarrow [0], l \leftarrow 1$.

for node $i \in N$, node $j \in N$ **do**
  for service $s \in S$ **do**
    if origin $o_s = i$ and destination $d_s = j$ **then**
      $p \leftarrow [s]$
      $P^l_{ij} \leftarrow P^l_{ij} \cup \{p\}$
      $MT^l_p \leftarrow$ AUXILIARYTIMEPOINTS(p)
$l \leftarrow l + 1$
while $l \leq L$ **do**
  for node $i \in N$, node $j \in N$ **do**
    for service $s \in S$ **do**
      if origin $o_s \neq i$ and destination $d_s = j$ **then**
        for feasible path $p \leftarrow [s_1, ..., s_{l-1}] \in P^{l-1}_{io_s}$ **do**
          if TIMECOMPATIBLE1$(p, s) = 1$ **then**
            $p' = [s_1, ..., s_{l-1}, s]$
            $P^l_{ij} \leftarrow P^l_{ij} \cup \{p'\}$
            $MT^l_{p'} \leftarrow$ AUXILIARYTIMEPOINTS(p')
  $l \leftarrow l + 1$

Algorithm 2. AUXILIARYTIMEPOINTS.

Input: Feasible path $p = [s_1, ..., s_l]$.
Output: Auxiliary time points $MT^l_p = \left[MT^{l1}_p, ..., MT^{l(2n)}_p, ..., MT^{l(2l)}_p\right], n \in \{1, ..., l\}$.
Initialize: Let $MT^l_p \leftarrow [0], n \leftarrow 1$.

while $n \leq l$ **do**
  if $s_n \in S^{\text{truck}}$ **then**
    if $n = 1$ **then**
      $MT^{l(2n-1)}_p \leftarrow 0$
    else
      $MT^{l(2n-1)}_p \leftarrow MT^{l(2n-2)}_p$
    Travel time of truck service $s_n \leftarrow$ calculate time-dependent travel time function of truck service s_n
    $MT^{l(2n)}_p \leftarrow MT^{l(2n-1)}_p$ plus loading time plus travel time of truck service s_n plus unloading time
  else
    $MT^{l(2n-1)}_p \leftarrow$ departure time of service $s_n \in S^{\text{barge}} \cup S^{\text{train}}$ minus loading time
    $MT^{l(2n)}_p \leftarrow$ arrival time of service $s_n \in S^{\text{barge}} \cup S^{\text{train}}$ plus unloading time
  $n \leftarrow n + 1$

Algorithm 3. TIMECOMPATIBLE1.

Input: node $i \in N$, service $s \in S \backslash S^+_i$, feasible path $p = [s_1, ..., s_{l-1}] \in P^{l-1}_{io_s}$.
Output: z, equal to 1 if path p and service s is time compatible, 0 otherwise.
Initialize: Let $z \leftarrow 0$.

if $s \in S^{\text{barge}} \cup S^{\text{train}}$ **then**
  if $MT^{2(l-1)}_p \leq$ departure time of service s minus loading time **then**
    $z \leftarrow 1$, return z
else
  $z \leftarrow 1$, return z

5.2.2. Preprocessing of feasible matches

A match $\langle r,p \rangle$ is defined as a combination of shipment $r \in R$ and path $p = [s_1, \ldots, s_l]$, $p \in P$, which means shipment r will be transported by the services included in path p. The match $\langle r,p \rangle$ is feasible only if it satisfies spatial and time compatibility: the origin of shipment r should be the same as the origin of service s_1, the destination of shipment r should be the same as the destination of service s_l; the release time of shipment r should be earlier than the virtual departure time point of service s_1.

We define Φ as the set of feasible matches, c_{rp} as the cost of matching shipment r with path p. Algorithm 4 is designed to create the feasible matches. For shipment r and path $p = [s_1, \ldots, s_l] \in P_{o_r d_r}{}^l$, the

time compatibility between r and p is checked by calling TIMECOMPATIBLE2, as presented in Algorithm 5. If $s_1, \ldots, s_n$ are truck services, the virtual departure and arrival time points of these truck services need to be updated sequentially. After the updating, if the virtual arrival time point of s_n is less than the virtual departure time point of service $s_{n+1} \in S^{\text{barge}} \cup S^{\text{train}}$, match $\langle r,p \rangle$ is feasible. If s_1 is a barge or train service, and the release time of shipment r is less than the virtual departure time point of service s_1, then match $\langle r,p \rangle$ is feasible.

Algorithm 4. Feasible match generation algorithm.

Input: Set of feasible paths P, set of shipment requests R, the largest number of services in a path L, index $l \in \{1, 2, ..., L\}$, set of auxiliary time points MT, objective function (1).
Output: Set of feasible matches $\Phi = \Phi_1 \cup \Phi_2 ... \cup \Phi_r ... \cup \Phi_R$.
Initialize: Let $\Phi \leftarrow \emptyset, l \leftarrow 1$.

```
for shipment request r ∈ R do
    for l ∈ {1, 2, ..., L} do
        for feasible path p = [s_1, s_2, ..., s_l] ∈ P^l_{o_r d_r} do
            if TIMECOMPATIBLE2(r, p) = 1 then
                Φ_r ← Φ_r ∪ {p}
                c_rp ← Calculate the objective function
```

Algorithm 5. TIMECOMPATIBLE2.

Input: shipment request $r \in R$, feasible path $p = [s_1, ..., s_l] \in P^l_{o_r d_r}$, auxiliary time points $MT^l_p = \left[MT^{l1}_p, ..., MT^{l(2n)}_p, ..., MT^{l(2l)}_p\right], n \in \{1, ..., l\}$.
Output: z, equal to 1 if r and p is time compatible, 0 otherwise.
Initialize: Let $z \leftarrow 0, n \leftarrow 2$.

```
if s_1 ∈ S^truck then
    update MT^{l1}_p ← release time of shipment request r
    update travel time of truck service s_1 ← calculate time-dependent travel time function truck service s_1
    update MT^{l2}_p ← MT^{l1}_p plus loading time plus travel time of truck service s_1 plus unloading time
    while n ≤ l do
        if s_n ∈ S^truck then
            update MT^{l(2n−1)}_p ← MT^{l(2n−2)}_p
            update travel time of truck service s_n ← calculate time-dependent travel time function truck service s_n
            update MT^{l(2n)}_p ← MT^{l(2n−1)}_p plus loading time plus travel time of truck service s_n plus unloading time
        else
            if MT^{l(2n−2)}_p ≤ MT^{l(2n−1)}_p then
                z ← 1, return z
            else
                return z
        n ← n + 1
else
    if release time of shipment request r ≤ MT^{l1}_p then
        z ← 1, return z
```

5.2.3. Binary integer programming

Based on the above preprocessing procedures, the objective function is updated to minimize the total costs for the matching of shipments with feasible paths. Let y_{rp} be a binary decision variable equal to 1 if shipment r is matched with path p, and 0 otherwise. The mathematical formulation translates into a binary integer programming (BIP) model:

Minimize

$$\sum_{r \in R} \sum_{p \in \Phi_r} c_{rp} y_{rp} \tag{30}$$

subject to

$$\sum_{p \in \Phi_r} y_{rp} = 1, \quad \forall\, r \in R, \tag{31}$$

$$\sum_{r \in R} \sum_{p \in \Phi_{rs}} y_{rp} q_r \le Q_s, \quad \forall\, s \in S^{\text{barge}} \cup S^{\text{train}}, \tag{32}$$

$$y_{rp} \in \{0, 1\}, \quad \forall\, r \in R, p \in \Phi_r, \tag{33}$$

where $\Phi_{rs} = \{p \in \Phi_r | s \in p\}$.

Constraints (31) ensure that only one feasible path will be assigned to each shipment. Constraints (32) ensure that the total volume of shipments assigned to service $s \in S^{\text{barge}} \cup S^{\text{train}}$ does not exceed its free capacity.

6. Numerical experiments

In this section, we first evaluate the performance of the optimization algorithms and compare the GA with the RHA. Then, we investigate the impact of different objective functions and optimization intervals. All algorithms were implemented in MATLAB R2017a, and all experiments were performed on a computer with 2.50 GHz Intel Core i5-7200U CPU and 8 GB RAM. CPLEX 12.6.3 was used as an IP solver.

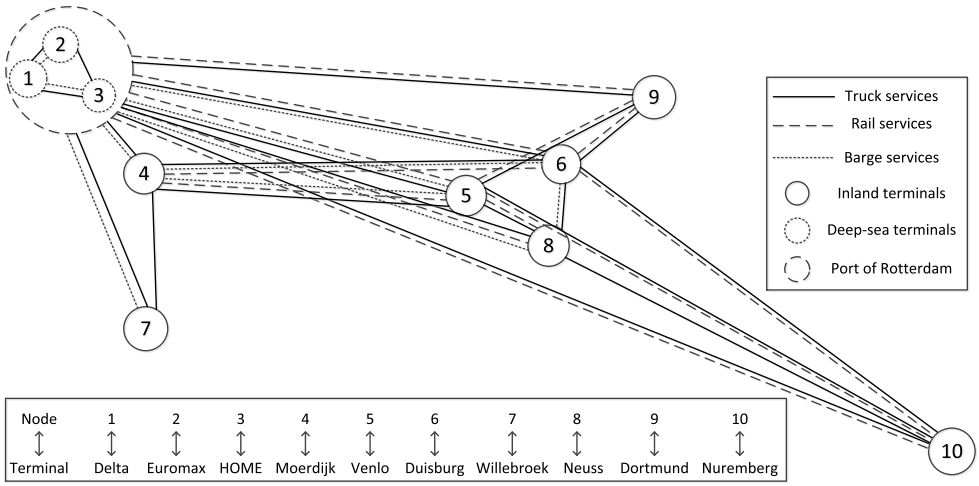

Fig. 6. The topology of an intermodal network in Europe.

6.1. Generation of test instances

In practice, different companies have different network sizes. For example, Combi Terminal Twente (https://www.ctt-twente.nl/en/, accessed: 2020-03-16) provides container transports from the port of Rotterdam to 3 inland terminals in the Netherlands and Germany with 7 barges, 3 trains and 40 trucks per week. European Gateway Services (EGS, https://www.europeangatewayservices.com/en, accessed: 2020-03-16) offers above 40 trains and 30 barges per week between the Ports of Rotterdam and Antwerp and 11 inland terminals in the Netherlands, Belgium, Germany, and Austria. Every year, approximately 1,000,000 TEU is transported within the EGS network. To show the application of the model, we consider a hinterland intermodal network in Europe to carry out the numerical experiments, as shown in Fig. 6. The network consists of three deep-sea terminals (nodes 1, 2, 3) and seven inland terminals (nodes 4, 5, 6, 7, 8, 9, 10) which are connected by 116 transport services, including 49 barges, 33 trains, and 34 trucks. The length of the planning horizon was set to one week. The coefficients used in the experiments were derived from Riessen et al. [20] and Li et al. [11], as shown in Table 2. Here, the transit cost of services is a linear function of the transit time t and distance d.

We generated several instances to represent different characteristics of shipments within the given network. We use EU $- n_1 - n_2$ to represent an instance with n_1 contractual requests and n_2 spot requests. The average container volume of contractual requests is 20 TEU, and the average container volume of spot requests is 5 TEU. We set the arrival frequency to 20, 10, 6 and 4 min for instances with 400, 800, 1200 and 1600 spot requests, respectively. Regarding the time-dependent travel times, we set $b_1 = 0$, $b_2 = 5$, $b_3 = 7$, $b_4 = 9$, $b_5 = 13$, $b_6 = 17$, $b_7 = 19$, $b_8 = 21$, $b_9 = 24$, $\alpha = 2$, $\beta = 1.5$. The detailed information of services and instances used in this paper is available at https://surfdrive.surf.nl/files/index.php/s/cCrpmO1dy8ls7if.

6.2. Performance of the heuristic algorithm

To compare the performance of the heuristic algorithm presented in Section 5.2 with the exact algorithm presented Section 5.1, we generated 8 instances of the DSM problem with different numbers of shipment requests. In the exact algorithm, we set the large enough number M to 168. In the heuristic setting, we let the largest number of services in a path L be 1, 2, 3 and 4, respectively. We use heuristic-L to represent the heuristic algorithm with setting L. The number of variables (i.e., N.var) and constraints (i.e., N.con) for the instances under different algorithms is presented in Table 3.

We consider two performance indicators: total costs (obj: €) and computation time (CPU: seconds). The computation time of heuristics includes the time of generating feasible matches and the time of solving the BIP model. We use 'gap' to represent the %gaps in total costs between different algorithms, which is given by (objective value − benchmark value) ∗ 100/benchmark value. Table 4 summarizes the performance for all instances. It shows that the small instances with up to 30 contractual requests are still solvable by using the exact algorithm. However, the computation time increases dramatically from 27 to 5647 s. In comparison, extending L from 1 to 3, the gaps in total cost between the heuristic algorithm and the exact algorithm decreases to 0.00%. The computation time of the heuristic algorithm with a maximum of 3 services in a path (Heuristic-3) is no more than 1 s.

For instances with above 700 total requests, we cannot obtain feasible solutions with the exact algorithm. The limitation in these instances is not the computation time but rather the memory since the size of the problems becomes too large to read. In contrast, all these large instances can be solved by using the heuristic algorithm with a maximum of 3 services in a path within 176.24 s, and the gaps in total costs between heuristic-3 and heuristic-4 are 0.00%.

6.3. Performance of the dynamic approaches

In this section, we aim to compare the performance of two dynamic approaches: the GA and the RHA. Both of them work with Heuristic-3. We set the length of the optimization interval under the RHA to 1 h.

We generated 4 groups of instances with different demand densities represented by the ratio between demand and supply: EU-100-400 (40%), EU-200-800 (80%), EU-300-1200 (120%), and EU-400-1600 (160%). Here, demand is the total container volumes of shipments, supply is the total free capacity of barge and train services. Each group includes 10 instances with the same ratio between demand and supply. We use the GA as the benchmark. Fig. 7(a) shows that the RHA has lower total costs in all the groups of instances, and the reduction in total costs increases with the demand density. The reason is that the higher the ratio between demand and supply, the competition between shipment requests is higher. The proposed RHA better allocates limited barge and train capacity to more suitable shipment requests which might arrive later in the system.

We generated another 4 groups of instances with different degrees of dynamism (DOD). In this paper, we define the DOD as the ratio between the number of spot containers and the number of total containers. Thus, the DOD for instance EU-300-400 is $(400 * 5)/(300 * 20 + 400 * 5) = 25\%$. The DOD for instance EU-300-400, EU-200-800, EU-100-1200, EU-0-1600 are therefore 25%, 50%, 75% and 100% respectively. Each group includes 10 instances with the same DOD. Fig. 7(b) shows that the RHA also has better performance in all

the groups of instances compared to the GA, and the improvement is increasing further with a higher DOD. Interestingly, when the matching system is 100% dynamic, the variance of the performance of the RHA becomes the largest. The reason is when the system is fully dynamic, the performance of the reoptimization-based RHA becomes uncertain.

To investigate the performance of the GA and the RHA under different lead time scenarios, we generated 3 groups of instances with different lead times of spot requests: EU-100-1200 (24), EU-100-1200 (48), and EU-100-1200 (72). Each group consists of 10 instances with the same lead time setting. Fig. 7(c) shows that the RHA has better performance than the GA in terms of total costs for all groups of instances and the improvement is larger for longer lead times. Longer lead times provide more flexibility for the RHA to re-optimize the decisions as new requests are received and the capacity can be allocated more effectively.

Similarly, we varied the response time of shipment requests from 1 h to 24 h for 3 groups of instances: EU-100-1200 (1), EU-100-1200 (12), and EU-100-1200 (24). Fig. 7(d) shows that the larger the response time, the better the performance of the RHA is in reducing total costs since it has more time to update decisions for all requests until their release times.

6.4. Impact of different objective functions and optimization intervals

In this section, we use the RHA and Heuristic-3 to investigate the impact of different objective functions and the length of the optimization interval.

6.4.1. Impact of different objective functions

We investigate the impact of different objective functions under instance EU-1000-0. The utilization of barges and trains is defined as the ratio between the utilized capacity of barge and train services multiplied by corresponding transit distances and the utilized total capacity of all services multiplied by corresponding distances. Table 5 shows that different objective functions generate different matching solutions. Comparing case 11 with cases 1 to 10, we observe that the total cost is the lowest when the objective function includes all elements. When we minimize the transit cost (case 1) or the carbon tax (case 5), the utilization of barges and trains is favored as they are cheaper and environmental friendlier than trucks. On the other hand, minimizing the transfer (case 2), storage (case 3) or delay (case 4) cost favors the utilization of trucks as they are faster in general and have flexible departure times. Comparing case 11 with cases 6 to 10, we see that the transit cost has the largest influence on the matching decisions while carbon tax has the smallest impact. However, it is predictable that the carbon tax coefficient will increase in the near future because of the increasing environmental issues and the enforced regulations. Under a restrict emission policy, such as case 14, including the carbon tax in the objective function can greatly affect the utilization of barges and trains. It is also interesting to observe that there is a clear trade-off between delay and carbon emissions as it is what is happening in real life.

6.4.2. Impact of the length of the optimization interval

To test the impact of the length of the optimization interval in the RHA, we used 4 instances with different DOD: EU-300-400 (25%), EU-

Table 2
Experimental setting.

Coefficient	Truck	Barge	Train
Transit cost (€/TEU-km-h)	30.98 t + 0.2758d	0.6122 t + 0.0213d	7.54 t + 0.0635d
Carbon emission (kg/TEU-km)	0.8866	0.2288	0.3146
Loading/unloading cost (€/TEU)	3	18	18
Loading/unloading time (h)	0	1	1
Carbon tax (€/ton)	8	8	8
Storage cost (€/TEU-h)	1	1	1

Table 3
Number of variables and constraints for the instances under different algorithms.

Instances	Exact algorithm		Heuristic-1		Heuristic-2		Heuristic-3		Heuristic-4	
	N.var	N.con	N.var	N.con	N.var	N.con	N.var	N.con	N.var	N.con
EU-5-0	4185	4221	26	18	54	25	66	25	68	25
EU-10-0	8370	8408	28	24	209	63	684	82	944	82
EU-20-0	16,740	16,676	84	61	428	85	1125	91	1488	91
EU-30-0	25,110	24,963	112	66	564	104	1646	105	2235	105
EU-700-0	585,900	580,996	2504	767	13,725	780	36,449	781	56,777	781
EU-1000-0	837,000	829,916	3279	1067	18,108	1082	49,908	1082	79,805	1082
EU-1300-0	1,088,100	1,079,016	4473	1367	25,377	1380	69,202	1381	109,758	1381
EU-1600-0	1,339,200	1,327,942	6032	1667	33,742	1680	91,020	1681	143,859	1681

Table 4
Performance of the heuristic algorithm with different L.

Instances	Exact algorithm		Heuristic-1		Heuristic-2		Heuristic-3		Heuristic-4		
	Obj	CPU	%gap	CPU	%gap	CPU	%gap	CPU	Obj	%gap	CPU
EU-5-0	4386	27.01	0.00	0.05	0.00	0.15	0.00	0.60	4386	0.00	0.28
EU-10-0	25,988	213.06	32.89	0.03	0.00	0.11	0.00	0.45	25,988	0.00	0.80
EU-20-0	44,198	1704.98	29.56	0.02	0.05	0.13	0.00	0.43	44,198	0.00	0.65
EU-30-0	65,126	5647.03	28.52	0.02	0.00	0.13	0.00	0.60	65,126	0.00	0.94
EU-700-0	Out of memory		17.49	1.37	0.17	8.21	0.00	25.47	1,060,077		38.43
EU-1000-0			18.37	2.60	0.25	16.46	0.00	45.22	1,017,669		78.94
EU-1300-0			19.03	6.12	0.42	34.15	0.00	94.62	1,042,481		158.57
EU-1600-0			18.36	10.55	0.17	63.22	0.00	176.24	1,020,075		302.41

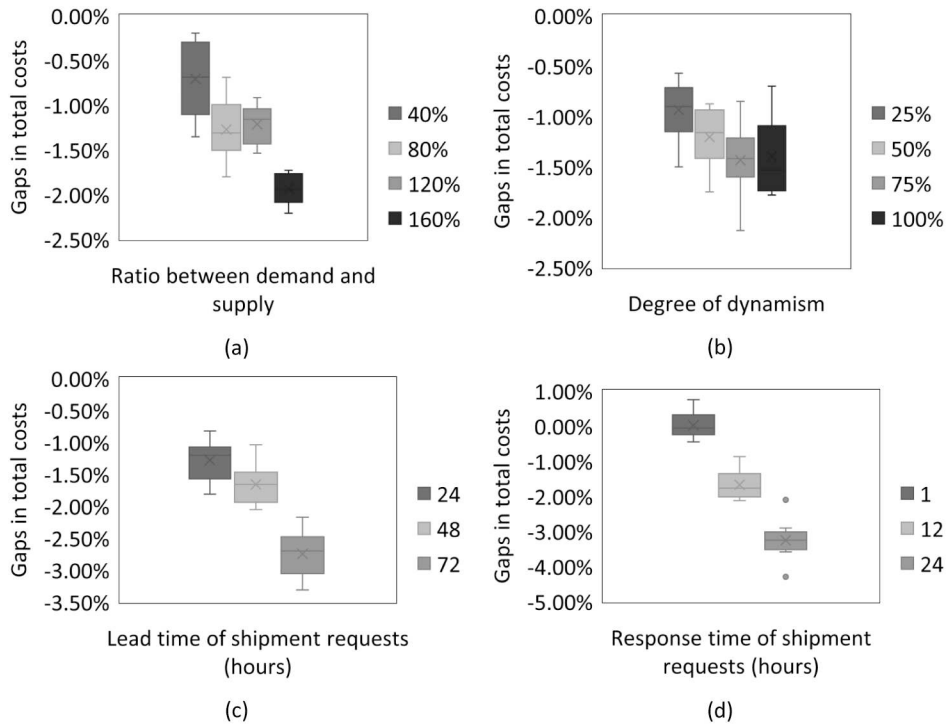

Fig. 7. Comparison between the rolling horizon approach and the greedy approach.

Table 5
Impact of different objective functions.

Case	Carbon tax coefficient (€/ton)	Objective function[a] (min.)	Total cost (€)	OF1 (€)	OF2 (€)	OF3 (€)	OF4 (€)	OF5 (€)	Delay (TEU-h)	Carbon emission (kg)	Utilization of barges and trains (%)	Utilization of trucks (%)
1	8	OF1	4,478,714	**598,864**	328,458	137,798	3,406,214	7379	39,170	922,429	71.47	28.53
2		OF2	1,473,382	1,411,229	**47,622**	0	0	14,530	0	1,816,311	0.00	100.00
3		OF3	1,618,747	1,499,961	103,374	**0**	0	15,412	0	1,926,482	0.04	99.96
4		OF4	1,617,409	1,495,824	105,960	245	**0**	15,379	0	1,922,413	0.42	99.58
5		OF5	4,432,293	601,621	324,498	144,863	3,353,948	**7364**	40,167	920,491	72.06	27.94
6		OF2,3,4,5	1,473,382	1,411,229	47,622	0	0	14,530	0	1,816,311	0.00	100.00
7		OF1,3,4,5	1,042,644	648,402	313,266	72,066	1112	7799	11	974,863	67.96	32.04
8		OF1,2,4,5	1,028,388	668,393	270,732	80,338	972	7953	10	994,084	65.78	34.22
9		OF1,2,3,5	1,803,565	656,501	260,772	69,829	808,619	7844	8624	980,454	66.71	33.29
10		OF1,2,3,4	1,017,693	695,156	252,702	60,783	880	8172	9	1,021,544	**63.76**	36.24
11		Total cost	**1,017,675**	692,118	254,448	62,114	850	8145	9	1,018,154	64.05	35.95
12	100	Total cost	1,110,869	684,140	260,790	64,039	972	100,929	10	1,009,287	64.78	35.22
13	500	Total cost	1,507,925	658,359	284,862	72,431	1162	491,111	12	982,222	66.94	33.06
14	1000	Total cost	1,995,063	643,700	298,386	78,945	8159	965,872	88	965,872	**68.48**	31.52

Bold to emphasis the significance of bold values.

[a] OF1: Transit cost; OF2: Transfer cost; OF3: Storage cost; OF4: Delay cost; OF5: Carbon tax; OF2,3,4,5: Transfer cost + Storage cost + Delay cost + Carbon tax; OF1,3,4,5: Transit cost + Storage cost + Delay cost + Carbon tax; OF1,2,4,5: Transit cost + Transfer cost + Delay cost + Carbon tax; OF1,2,3,5: Transit cost + Transfer cost + Storage cost + Carbon tax; OF1,2,3,4: Transit cost + Transfer cost + Storage cost + Delay cost.

200-800 (50%), EU-100-1200 (75%) and EU-0-1600 (100%). For each instance, we vary the length of the optimization interval h from 0.1 to 10 h.

We use optimization intervals of 1 h as the benchmark. Fig. 8 shows that reducing h allows the system to react more quickly to new information, which in turn leads to improved solutions. This is especially the case for instances with a high DOD. However, excessively reducing h does not improve the performance of the RHA. It is seen that below 1 h of optimization intervals does not bring values as expected since the response times are set as a minimum of 1 h. Therefore, decision makers can improve the matching quality by choosing a proper h-value.

7. Conclusion and future research

In this paper, we introduced an online synchromodal matching problem in which a platform aims to provide optimal matches between shipment requests and transport services. We proposed a rolling horizon approach and a heuristic algorithm to support the online decision-making process. We validated the heuristic algorithm and the rolling horizon approach on an intermodal network in Europe. The results indicate that the heuristic algorithm is efficient in large instances of the matching problem, and can be used under dynamic contexts. The rolling horizon approach has been proved to outperform a greedy approach in reducing total costs under various scenarios.

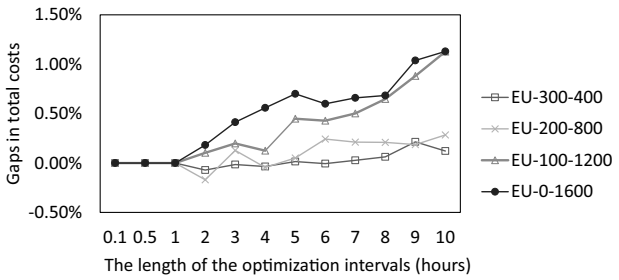

Fig. 8. Impact of the length of the optimization interval.

In conclusion, the proposed online matching platform will support decision makers to optimize the matching of shipments and services considering the trade-off between transport cost, delay, and carbon emissions thanks to the developed rolling horizon approach. In other words, with the proposed approach, the use of barges, trains, and trucks can be managed more effectively taking into account their impact on transport time, cost and emissions together with different time sensitivities of shipments.

This work can be extended in several directions. During the day, the number of trucks available to the matching platform is quite dynamic. Therefore, combining the dynamics of truck services in the synchromodal matching model is a further research direction. Considering the multiple uncertainties that exist in synchromodal transportation, future research can be carried out on *stochastic and dynamic* shipment matching. Furthermore, the origins and destinations of containers are usually located in different countries. Thus, looking into models with an *integrated* network combining international and inland transport is a promising research direction. Besides, in this paper, the online matching platform is controlled in a centralized way. However, in practice, multiple operators are present and they may not all be willing to give authority to a central platform. The *coordination mechanism* among them and *incentives* to stimulate cooperation are part of future research.

CRediT authorship contribution statement

Wenjing Guo:Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing - original draft, Writing - review & editing, Visualization.**Bilge Atasoy**:Conceptualization, Formal analysis, Visualization, Writing - review & editing, Supervision.**Wouter Beelaerts van Blokland**:Conceptualization, Writing - review & editing, Supervision.**Rudy R. Negenborn**:Conceptualization, Writing - review & editing, Supervision.

Acknowledgments

This research is financially supported by the China Scholarship Council under Grant 201606950003 and the project “Complexity Methods for Predictive Synchromodality” (project 439.16.120) of the Netherlands Organisation for Scientific Research (NWO).

References

[1] T. Ambra, A. Caris, C. Macharis, Towards freight transport system unification: reviewing and combining the advancements in the physical internet and synchromodal transport research, International Journal of Production Research 57 (6) (2018) 1606–1623.
[2] A.M. Arslan, N. Agatz, L. Kroon, R. Zuidwijk, Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers, Transportation Science 53 (1) (2019) 222–235.
[3] A. Bhattacharya, S.A. Kumar, M. Tiwari, S. Talluri, An intermodal freight transport system for optimal supply chain logistics, Transportation Research Part C: Emerging Technologies 38 (2014) 73–84.
[4] T.G. Crainic, P. Dell'Olmo, N. Ricciardi, A. Sgalambro, Modeling dry-port-based freight distribution planning, Transportation Research Part C: Emerging Technologies 55 (2015) 518–534.
[5] T.G. Crainic, K.H. Kim, Chapter 8 intermodal transportatio, Transportation, Elsevier, 2007, pp. 467–537.
[6] E. Demir, W. Burgholzer, M. Hrušovský, E. Arıkan, W. Jammernegg, T. Van Woensel, A green intermodal service network design problem with travel time uncertainty, Transportation Research Part B: Methodological 93 (2016) 789–807.
[7] E. Demir, M. Hrušovský, W. Jammernegg, T. Van Woensel, Green intermodal freight transportation: bi-objective modelling and analysis, International Journal of Production Research 57 (19) (2019) 6162–6180.
[8] S. Fazi, J.C. Fransoo, T. Van Woensel, A decision support system tool for the transportation by barge of import containers: a case study, Decision Support Systems 79 (2015) 33–45.
[9] R. Giusti, D. Manerba, G. Bruno, R. Tadei, Synchromodal logistics: an overview of critical success factors, enabling technologies, and open research issues, Transportation Research Part E: Logistics and Transportation Review 129 (2019) 92–110.
[10] R. Konings, E. Kreutzberger, V. Maraš, Major considerations in developing a hub-and-spoke network to improve the cost performance of container barge transport in the hinterland: the case of the port of Rotterdam, Journal of Transport Geography 29 (2013) 63–73.
[11] L. Li, R.R. Negenborn, B. De Schutter, Intermodal freight transport planning – a receding horizon control approach, Transportation Research Part C: Emerging Technologies 60 (2015) 77–95.
[12] M.R.K. Mes, M.E. Iacob, Synchromodal transport planning at a logistics service provider, Logistics and Supply Chain Innovation, Springer International Publishing, 2015, pp. 23–36.
[13] A. Najmi, D. Rey, T.H. Rashidi, Novel dynamic formulations for real-time ride-sharing systems, Transportation Research Part E: Logistics and Transportation Review 108 (2017) 122–140.
[14] V. Pillac, M. Gendreau, C. Guéret, A.L. Medaglia, A review of dynamic vehicle routing problems, European Journal of Operational Research 225 (1) (2013) 1–11.
[15] W. Qu, J. Rezaei, Y. Maknoon, L. Tavasszy, Hinterland freight transportation replanning model under the framework of synchromodality, Transportation Research Part E: Logistics and Transportation Review 131 (2019) 308–328.
[16] A.P. Rivera, M. Mes, Service and transfer selection for freights in a synchromodal network, Lecture Notes in Computer Science, Springer International Publishing, 2016, pp. 227–242.
[17] M. SteadieSeifi, N. Dellaert, W. Nuijten, T. Van Woensel, R. Raoufi, Multimodal freight transportation planning: a literature review, European Journal of Operational Research 233 (1) (2014) 1–15.
[18] P. Sun, L.P. Veelenturf, M. Hewitt, T. Van Woensel, The time-dependent pickup and delivery problem with time windows, Transportation Research Part B: Methodological 116 (2018) 1–24.
[19] B. van Riessen, R.R. Negenborn, R. Dekker, Real-time container transport planning with decision trees based on offline obtained optimal solutions, Decision Support Systems 89 (2016) 1–16.
[20] B. van Riessen, R.R. Negenborn, R. Dekker, G. Lodewijks, 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) (2015) 457.
[21] H. Wang, X. Wang, X. Zhang, Dynamic resource allocation for intermodal freight transportation with network effects: approximations and algorithms, Transportation Research Part B: Methodological 99 (2017) 83–112.
[22] X. Wang, Optimal allocation of limited and random network resources to discrete stochastic demands for standardized cargo transportation networks, Transportation Research Part B: Methodological 91 (2016) 310–331.
[23] R.A. Zuidwijk, A.W. Veenstra, The value of information in container transport, Transportation Science 49 (3) (2015) 675–685.

Wenjing Guo is a PhD researcher at Delft University of Technology. Her research interests include operations research, intermodal transportation, dynamic optimization, stochastic optimization, and distributed optimization. Her main ambition is to combine advanced approaches with practical applications considering the trend towards sustainability, ecommerce, and digitalization in freight transportation. She received her MSc degree from Wuhan University of Technology in 2013 in the area of agent-based negotiation strategies in agricultural supply chains which received the excellent Master Thesis Award.

Bilge Atasoy is an assistant professor at TU Delft within the Department of Maritime and Transport Technology. Her main research interests lie at the intersection of optimization and behavioral models. She applies the scientific methodologies in the field of various transport and logistics problems in order to increase sustainability and efficiency. One of the application areas is transport and logistics over water where fleet management models are developed to optimize several decisions (e.g., fleet size, needed capacity, routes and schedules for vessels). Prior to joining TU Delft, Bilge was a research scientist at MIT at the Intelligent Transportation Systems (ITS) Lab where she is now a research affiliate. At MIT she led several research projects in the areas of real-time optimization, travel behavior and choice-based optimization. Bilge obtained her PhD from EPFL in November 2013 in the area of integrated supply and demand models in transportation problems which received the best PhD Thesis Award from the Swiss Operations Research Society. She received her MSc and BSc degrees in Industrial Engineering from Bogazici University, Istanbul, in 2009 and 2007, respectively.

Wouter Beelaerts van Blokland researches theories supporting Advanced Operations

and Production Management such as Lean manufacturing, value chain and system, (maintenance) supply chains and value creation by innovation. Central theme is performance measurement with KPI's regarding the flow of components or sub systems through processes, to support the process performance regarding the coordination of assets and resources. Currently he is assistant professor at Delft University of Technology within the section Multi-Machine Engineering, part of the department Marine & Transport Technology of the faculty of Mechanical, Marine and Materials engineering. He achieved his PhD at the Delft University of Technology in 2010. After working in the drive and control and aerospace industry he started this PhD in 2004 with the faculty of Aerospace Engineering to research the effect of leveraging value on suppliers by aircraft manufacturers for the co-development and co-production of aircraft. In that time he started to lecture on Lean Operations Performance Assessment and Value Engineering from which several start-up companies were initiated such as "Type22" and Fly Aeolus. He was nominated for the Delft Entrepreneurial Scientist Award 2010, category Entrepreneurial Motivator and principal lecturer to the team on the project "Formation Flyer", which won the National Prize on aeronautics (2011) in the Netherlands.

Rudy R. Negenborn is a full professor in Multi-Machine Operations & Logistics. He is head of the Section Transport Engineering & Logistics of Department Maritime & Transport Technology. His research interests include intelligent infrastructures & logistics, decision making and coordination for transport technology (including smart vessels) in general, whereby he proposes multi-agent system and model predictive control approaches that benefit from real-time information availability and the potential of communication. As such, his research anticipates the massive introduction of sensing, computation, and communication technologies. This is materialized into innovative solutions for smart equipment, transport hubs, ports and (synchromodal) transport networks. He has over 200 peer reviewed academic publications. He leads NWO, EU and industry funded research, and is on the editorial board of the series on "Intelligent Systems, Control and Automation: Science and Engineering". He was moreover general chair of the 6th International Conference on Computational Logistics, has acted as member of the organizing committee of several other international conferences (including IEEE control conferences and maritime systems & logistics conferences) and was guest editor of special journal issues on autonomous vessels and computational logistics. In addition, he is the editor of the books "Intelligent Infrastructures", "Distributed Model Predictive Control Made Easy", and "Transport of Water versus Transport over Water".