The Multi-Port Berth Allocation Problem with Speed Optimization: Exact Methods and a Cooperative Game Analysis

Bernardo Martin-Iradi

DTU Management, Technical University of Denmark, Akademivej Building 358, 2800 Kgs. Lyngby, Denmark, bmair@dtu.dk

Dario Pacino

DTU Management, Technical University of Denmark, Akademivej Building 358, 2800 Kgs. Lyngby, Denmark, darpa@dtu.dk

Stefan Ropke

DTU Management, Technical University of Denmark, Akademivej Building 358, 2800 Kgs. Lyngby, Denmark, ropke@dtu.dk

We consider a variant of the berth allocation problem —i.e., the multi-port berth allocation problem—aimed at assigning berthing times and positions to vessels in container terminals. This variant involves optimizing vessel travel speeds between multiple ports, thereby exploiting the potentials of a collaboration between carriers (shipping lines) and terminal operators. Using a graph representation of the problem, we reformulate an existing mixed-integer problem into a generalized set partitioning problem, in which each variable refers to a sequence of feasible berths in the ports that the vessel visits. By integrating column generation and cut separation in a branch-and-cut-and-price procedure, our proposed method is able to outperform commercial solvers in a set of benchmark instances and adapt better to larger instances. In addition, we apply cooperative game theory methods to efficiently distribute the savings resulting from a potential collaboration and show that both carriers and terminal operators would benefit from collaborating.

Key words: Transportation, Exact methods, Container terminal, Berth allocation problem, Speed Optimization, Cooperative game theory

1. Introduction

The International Maritime Organization (IMO), in its fourth climate report (IMO 2020), reflects on the increase in shipping's CO_2 emissions in the recent years. In the period 2012-2018 the shipping's total emissions have increased by 9.6%. This alarming trend highlights the need for pursuing the strategies that the IMO adopted in 2018 for reducing greenhouse gas (GHG) emissions from ships (IMO 2018). The aim is to reduce total emissions from shipping by 50% in 2050, and to reduce the average carbon intensity by 40% in 2030 and 70% in 2050, compared to 2008. Yet world maritime trade keeps growing at an annual average of 3% reaching a record high of 11 billion tons of total volume in 2018 —a number that translates into almost 800 million twenty-foot equivalent units (TEUs) handled in container ports worldwide (UNCTAD 2019). Given that trade volume has

steadily increased since then, the need for more efficient and sustainable operations in maritime transport logistics is essential (Bektaş et al. 2019).

From the terminal viewpoint, the growth in container trade involves more or larger vessels arriving at ports, in need of berthing. One solution to satisfying the increasing demand is to extend the existing quay. The problem is that doing so usually requires an expensive investment and sometimes may not even be physically feasible. An alternative strategy is to improve the efficiency of existing resources through optimization techniques that do not entail costly investment.

The berth planning of a terminal can be modelled mathematically as the Berth Allocation Problem (BAP). In the BAP, the aim is to assign incoming ships to berthing positions along the terminal. Steenken, Voß, and Stahlbock (2004) define this problem as highly critical within container container terminal planning logistics, due to the scarcity of berthing space. Figure 1 illustrates the
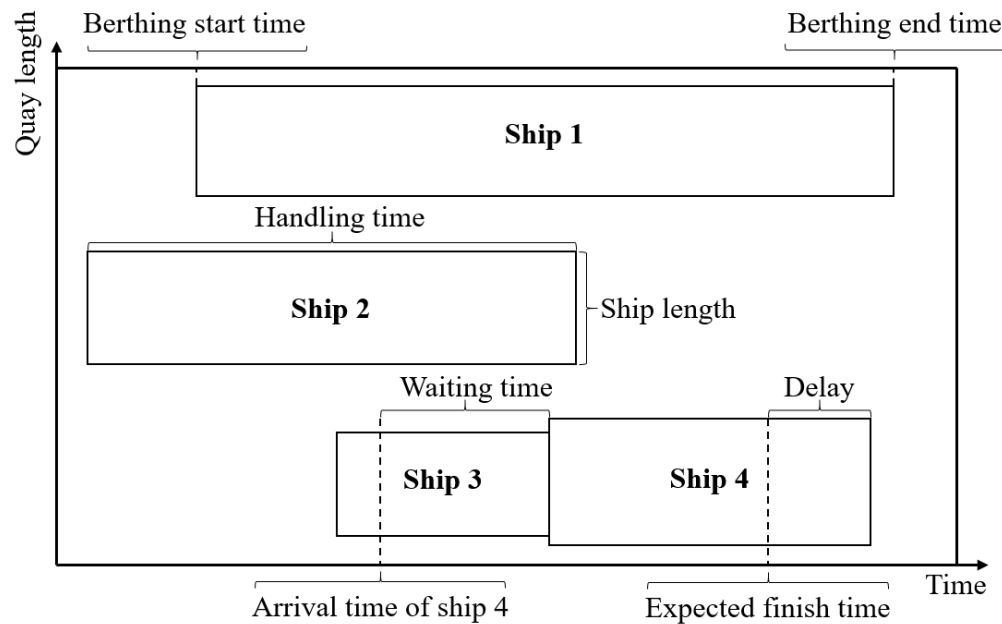

Figure 1 **Example solution of the BAP for a port terminal with four vessels.**

problem in a two-dimensional diagram where one dimension is space (quay length), and the other one is time (the planning horizon). We depict each ship as a rectangle whose dimensions are the ship length and handling time, the time the ship spends at the berth (i.e., during unloading and loading) Each ship usually has a fixed time window defined by its expected start and finish time. Although ships can arrive before their berthing time, they will need to wait at the port. Similarly, ships can be allowed to exceed the expected finish time incurring in a delay. We denote the entire time that the ship spends at the port (i.e waiting time plus berthing period) as the "service time." Any non-overlapping positioning of the ship rectangles within the decision space defines a feasible solution for the BAP.

We can classify the BAP variants according to how the berths are distributed along the quay. In the *discrete* BAP, we divide the quay into a discrete set of berths, with only one ship allowed to one berth at a time, whereas in the *continuous* BAP, the ships can berth anywhere along the quay as long as they maintain a safe distance from the other ships. Moreover, the BAP can be either *static* or *dynamic*. In the *static* variant, we assume that all ships are at the port when the berth planning is done, whereas in the *dynamic* version, we assume that ships can arrive while the planning is in process.

The efficient planning of a terminal requires the vessels to abide by their schedules. Thus, efficient vessel scheduling is also a critical aspect, not only for the carriers, but also for the terminal operators. The design of vessel schedules can be modeled mathematically as the Vessel Scheduling Problem (VSP). The VSP aims at determining the sailing speeds between consecutive ports in the route (i.e., voyage legs) in order to optimize the vessels' fuel consumption and turnaround time at port and the number of vessels required to operate the route with a given frequency.

Both the increasing volume of container trade and the up-sizing of the vessels have led to increased competition among container terminals. each vying to become the port of call for more vessels (Notteboom et al. 2017). As a result, most terminals are reticent to share information with other terminals and prefer to plan their operations independently. Terminals commonly plan berth allocation based on ship schedules. Nevertheless, these schedules are subject to a level of uncertainty, because different types of disruptions—such as weather conditions or technical problems at the terminal—can alter the schedules and result in delays. When each terminal does its planning independently, a delay in one terminal can potentially be propagated through the shipping service to other ports (Notteboom and Vernimmen 2009) or incur higher fuel costs for the carriers (shipping lines) if they need to increase the vessel's speed to make up for lost time. For example, a vessel stopping in ports A and B may encounter a congested terminal when arriving at port A and become delayed. The carrier can then order the vessel to either speed up to arrive at port B on time, entailing higher fuel consumption, or arrive late at port B, forcing the terminal to modify its berthing plan.

A potential solution to avoid this type of scenario is to establish some form of collaboration between players in the maritime industry. Collaboration can be established not only between same type of stakeholders (i.e., between multiple carriers) but also between more players (i.e., carriers and terminal operators). The World Shipping Council (2015) encourages terminals to establish collaborative agreements with carriers as one of the main ways of reducing port congestion and improving planning efficiency.

A certain degree of collaboration is assumed in the VSP, however, the problem does not explicitly consider the berth allocation at the terminal and this can lead to a significant increase in service

4

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

time. Therefore, integrating the BAP together with the scheduling of the vessels becomes relevant. Sharing information allows planners to simultaneously plan the berthing at the terminals and be able to minimize disruptions and reduce costs and emissions. Recent studies show that collaboration between carriers and terminals can lead to significant benefits for both (Dulebenets et al. 2019). This is the goal of the Multi-Port Berth Allocation Problem (MPBAP), first introduced by Venturini et al. (2017), which simultaneously plans the berth allocation of multiple ports taking into account the vessels' speed.

The MPBAP can either be applied either when one company controls both vessels and terminals, or by a third-party service provider which works as an orchestrator. An example of the former is *Maersk*, owning both the carrier *Maersk Line* (Maersk 2021) and the terminal operator *APM Terminals* (APM Terminals 2021).

At present, there are companies in the market that offer optimization-based planning software separately to carriers and terminal operators (Portchain 2021, Navis 2021, Sealytix 2021, TGI 2021, RBS 2021). Such companies already have access to all the necessary data for the MPBAP, which makes them excellent candidates to orchestrate the collaboration. Since both carriers and terminals are already sharing data with those companies, trust issues should be minimal, but customers should of course be free to decline that their data is used in a joint optimization problem. The amount of flexibility that terminals and carriers are willing to commit to the collaboration, can easily be modeled with the time windows, making the MPBAP if not an operational tool, at least a tool to identify the potential savings.

To make the service attractive to customers, the software company needs to show that the collaboration is beneficial for all involved parties. Therefore, we apply cooperative game theory to demonstrate that the total costs in the MPBAP solution can be shared in a favorable way. Using this service only requires that participating carriers and terminal operators allow the third party to jointly use their data but does not entail sharing additional data or the disclosure of the customer's data to other customers. Once the operations conclude, the third party would be in charge of returning the savings according to the initial calculations.

Similar collaboration mechanisms have also been studied in the road transportation sector. Ergun, Kuyzu, and Savelsbergh (2007) study collaborative logistics in truck transportation where part of the carriers' savings are returned to the shippers. Özener, Ergun, and Savelsbergh (2011) also propose collaborative models where players receive more favorable rates in return. In fact, they indicate that a centralized decision-maker with complete information about all participants would be ideal for collaborative models to work. However, in their study, Özener, Ergun, and Savelsbergh (2011) suggest that, due to lack of trust, players may not be willing to share additional information

and therefore, they explore different mechanisms. Fortunately, this lack of trust is minimized in our case as the players already share the required information with the third party.

In studying the MPBAP, this paper makes four contributions. First, we present two new formulations for the MPBAP, based on a graph representation. Second, we propose exact methods based on column generation, together with branching, cutting, and symmetry-breaking enhancements. Third, we demonstrate the quality of our method by comparing it to a commercial solver and testing it through both a set of benchmark instances from a previous study and a new set of harder instances. Fourth, to demonstrate the benefits for both carriers and terminal operators in a scenario of a joint grand coalition, we apply cost allocation methods from cooperative game theory.

The structure of this paper is as follows. Section 2 reviews the state-of-the-art studies on berth allocation, speed optimization and collaboration on the shipping industry. Section 3 describes the MPBAP by presenting two mathematical formulations, together with the one from Venturini et al. (2017). Section 4 gives our solution method, and Section 5 introduces and discusses the cooperative game methods used for effectively distributing the costs of a coalition. Section 6 compares the models' performance through extensive computational experiments and analyzes the cooperative game theory results. Section 7 concludes by briefly discussing both the findings and possible future research directions.

2. Literature review

This section has been divided into three. First, we describe the main studies related to the BAP. Secondly, we cover the literature concerning speed optimization, and the last part focuses on collaboration studies within the container shipping industry and literature where cooperative game theory has been applied to it.

2.1. BAP literature

The berth allocation problem is known to be NP-hard (Lim 1998, Hansen and Oguz 2003) and has received significant attention in the last two decades. Carlo, Vis, and Roodbergen (2014) and Bierwirth and Meisel (2015) presented detailed literature surveys on the seaside operations of container terminals such as the BAP where they emphasized the raising interest on this particular problem in the last years. Imai et al. (2005) conducted the first study considering a continuous BAP and Cordeau et al. (2005) studied both the discrete and continuous version of the problem and solved them through heuristic methods. Guan and Cheung (2005) presented a tree search exact method that performed better than commercial solvers and an efficient composite heuristic method. Du et al. (2015) extended the problem to also include the effect of tides and adopted the *virtual arrival* policy that is currently used in many terminals worldwide. Cheong et al. (2010)

considered priorities for each of the vessels. The BAP is optimized using an evolutionary algorithm that minimizes the make-span, the waiting time and the deviation from a reference schedule. Buhrkal et al. (2011) compared three different methods for the discrete BAP and showed that a generalized set-partitioning model outperforms the rest. Saadaoui, Umang, and Frejinger (2015) reformulated the problem into a set packing problem where variables refer to assignments of ships to berthing positions and solved it using delayed column generation. In our paper, we combine the applicability of column generation procedures using a generalized set partitioning problem formulation. Regarding the discretization of the quay, Kordić et al. (2016) presented a hybrid variant of the BAP where ships can only berth in a given set of positions. Lalla-Ruiz et al. (2016) studied how the tides can limit the time available for ships to berth given their draft and the water depth and solved this variant of the BAP using a generalized set partitioning problem formulation. The multi-port version of the BAP studied in this paper was first defined by Venturini et al. (2017). The mixed integer problem formulation they presented is used as a reference for the ones considered in this paper. Kramer et al. (2019) proposed two new formulations for the discrete BAP: a time-indexed formulation and an arc-flow formulation that seem to perform better than the methods from Buhrkal et al. (2011). Corry and Bierwirth (2019) proposed a mixed integer problem formulation for the BAP with channel-constrained ports where the sequencing of channel movements is also optimized.

2.2. Speed optimization literature

The relation between vessel speed and fuel consumption is non-linear. Since fuel emissions are directly proportional to the fuel burnt, optimizing sailing speed becomes relevant from the carrier and environmental perspective. The policies of the IMO in the last years have raised debate on which measures to implement regarding speed optimization, speed reduction or slow steaming. In that aspect, multiple studies have been done analyzing the aspects and impacts of the different measures. Based on the scenario of slow steaming, Kontovas and Psaraftis (2011) investigated a berthing policy that aims at reducing the waiting time at port. Psaraftis and Kontovas (2013), Wang, Meng, and Liu (2013a), Psaraftis and Kontovas (2015a) and Psaraftis and Kontovas (2015b) presented taxonomies and surveys on speed models in the maritime transportation sector where the impacts and main trade-offs of slow steaming are analyzed and decision models proposed.

The VSP has speed optimization as its core concept and the interest in this problem has continued increasing in the last decade (Dulebenets et al. 2019). Fagerholt (2001) presented a mathematical model for the VSP and solved it using a method based on the set partitioning formulation. Wang, Wang, and Meng (2014) extended the VSP to also consider cargo allocation and indicated that carriers should consider the cargo costs arising from additional waiting time at port. Dulebenets

(2018) proposed a multi-objective model considering the route service costs. The results indicated that negotiating the port calls and handling rates with the terminal operator could lead to significant savings. To some extent, the VSP can be seen as a collaborative problem, however, most of the studies focus on the interests of the carrier. A variant of the VSP where shipping line companies and terminal operators collaborate has also been studied recently. This variant assumes that the terminal operator can offer multiple time windows or handling rates to the carrier, instead of the fixed ones considered in the generic VSP. For instance, the MPBAP presented in Venturini et al. (2017) can be included in this problem category where the berth allocation planning is also considered. Dulebenets (2019) presented a mathematical model for the collaborative VSP where terminals offer both multiple port service time windows and handling rates. The results showed the benefits of the collaborative agreement on the liner shipping operations.

Environmental aspects have also been addressed in this type of problems. Fagerholt, Laporte, and Norstad (2010) minimized the fuel consumption by optimizing speeds along a shipping route. By discretizing the arrival times at each port, the cubic function relating speed and fuel emissions can be linearized and the problem solved as a shortest path problem. Fagerholt et al. (2015) and Zhen et al. (2020) extended the route and speed optimization study by also considering emission control areas (ECAs). Fagerholt et al. (2015) aimed at minimizing the fuel consumption whereas Zhen et al. (2020) also considered SO_2 emissions. Both studies showed that carriers tend to use slow steaming within ECAs or directly avoid sailing through these areas. Reinhardt et al. (2016) optimized a liner shipping network by adjusting berthing times with the objective of minimizing fuel consumption. The speed and routing of multiple vessels is optimized in Wen, Pacino, and Kontovas (2017) under a unified objective that minimizes transit times, total costs and fuel emissions. They implemented a branch-and-price heuristic and a constraint programming model which is tested in a subset of the Mediterranean ports. Du et al. (2011), Du et al. (2015) and Sun et al. (2018) integrated speed optimization with the BAP by considering that ships still need to sail a certain distance to arrive at port. The second-order cone programming transformation used by Du et al. (2011) to approximate the relation between sailing speed and fuel consumption is improved by quadratic outer approximations in Wang, Meng, and Liu (2013b).

2.3. Collaboration in the shipping industry

The MPBAP introduced by Venturini et al. (2017) can be seen as a problem with a high degree of collaboration and the study of different collaborative forms in the container shipping industry has gained interest in recent years. Song (2003) studied competition and co-operation in ports and coined the term *co-opetition*. Wang, Liu, and Qu (2015) presented two collaborative methods between shipping line companies and port operators where the aim is to create a win-win situation

by balancing the priorities of both parties and encouraging them to share true information. Lalla-Ruiz, Melián-Batista, and Moreno-Vega (2016) proposed a cooperative search for the discrete BAP based on a grouping strategy. Individuals are organized into groups where they can only share information with other individuals from the same group. Notteboom et al. (2017) investigated alliance formations in container shipping by studying their strategies when choosing ports. Dulebenets, Golias, and Mishra (2018) presented the collaborative berth allocation problem (CBAP), which is a variation of the BAP that also allows to divert vessels to another terminal when there is a peak demand, and solved it using a memetic algorithm. Collaboration is also studied by integrating berth allocation with other scheduling problems such as ship routing. Pang and Liu (2014) studied such integration for a feeder company operating both vessels and container terminals. This study also considered transhipments of containers but did not cover speed optimization.

Game theory has also been widely applied in the container shipping industry (Pujats, Golias, and Konur 2020). In our paper, the focus is on cooperative game theory where the target is on distributing the profits or savings among players. The studies vary depending on which are the players considered (carriers, terminal operators or both) in the cooperation. Song and Panayides (2002) applied cooperative game theory to depict a conceptual framework for liner shipping alliances showing that the core theory is applicable to the liner shipping market. Saeed and Larsen (2010) presented a two-stage cooperative game for container terminals within the Karachi Port in Pakistan. The results indicated that a *grand coalition* among all players gives the best payoff for all terminals. The work by Krajewska et al. (2008) showed, by means of cooperative game theory, that collaboration among road freight carriers is practical and cost-effective for all players. Wen et al. (2019) studied the benefits of horizontal cooperation in a shipping pool by not only maximizing the pool profit but also allocating the profits fairly among participants. The profit sharing framework from Krajewska et al. (2008) and some of the profit allocation methods presented in Wen et al. (2019) have been used in this study and, to the best of our knowledge, it is the first time cooperative game theory is applied to the MPBAP.

2.4. Research gap

While the BAP and VSP have been extensively studied in the literature, with an increasing interest in the last decade, very few papers address the potentials of integrating the two problems and only one paper has been found to address the MPBAP. Only an MIP formulation for the problem has been proposed, which shows good performance for small instances but struggles when the size of the instances increases. Therefore, there is a need for a more efficient solution method that scales better to larger instances. Furthermore, the MPBAP implies collaboration between different parties in the shipping industry and an analysis of the model's applicability in real life is lacking in the literature. Thus, we believe that assessing the stakeholders' incentives to enter into such collaboration is relevant.

3. Problem description

The MPBAP can be seen as a partial integration between the BAP and the VSP. Particularly, this study is based on the discrete and dynamic BAP and it is extended to cover multiple ports where the sailing speed between ports is optimized. This can be seen as a collaborative approach where information is shared among shipping line and terminal companies. The main addition of the MPBAP compared to the BAP is the optimization of the sailing speed between ports and the simultaneous planning of multiple ports. Figure 2 shows a solution example to a problem with four ships and two ports, each having three berthing positions. As shown for ship 1, the travel time, which depends on the chosen sailing speed, determines the arrival time to the next port and this can constrain the available berthing time window further. The MPBAP aims at minimizing the total costs for both the carriers and terminal operators.
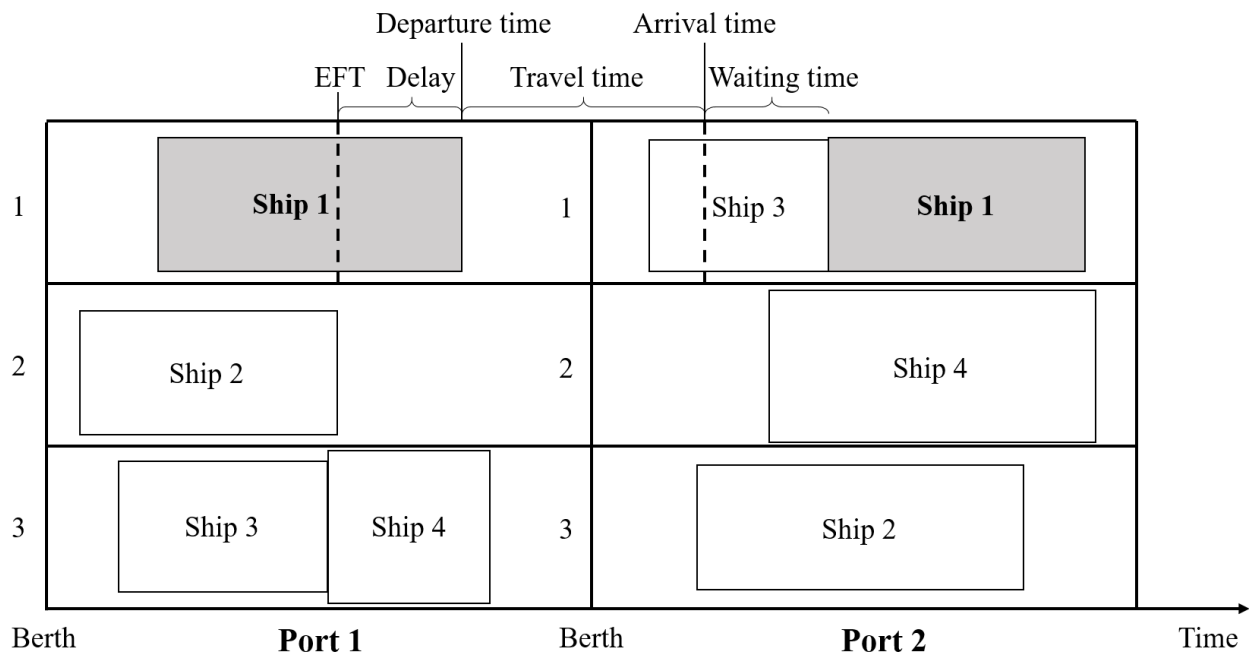

Figure 2   **Example solution of the MPBAP for four vessels, two port terminals and three berths per port. The traveling timeline for ship 1 (in gray) is defined at the top, where** EFT **denotes the expected finish time at port 1.**

3.1. Fuel consumption model

One of the main costs for a carrier is the fuel. The fuel consumption is directly linked to the sailing speed but not in a linear way. Thus, we need an accurate model that links the sailing speed with the fuel consumption realistically.

10

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

Many studies approximate the fuel consumption as a cubic function of the speed (e.g., Meng and Wang (2011), Wang and Meng (2012), Reinhardt et al. (2016)),

$$F(i,\delta) = \left(\frac{\delta}{\delta_i}\right)^3 \Gamma_i \tag{1}$$

where equation (1) measures the fuel consumption $F(i,\delta)$ in tons/hour for a given ship i. δ_i is the design speed of vessel i and δ is the sailing speed, both measured in knots (i.e., nautical miles per hour). Finally, Γ_i is the fuel consumption in tons/hour for vessel i at the design speed. This approximation is fairly accurate for container ships of limited size and for a range of sailing speeds that are not significantly slow. In our study, we optimize the sailing speed between ports, where we expect speeds similar to the design speed (δ_i) of the vessel and we do not consider the fuel consumption derived from entering or leaving a port where near-zero speeds are used. In order to avoid non-linearity in the mathematical formulation of the problem, we apply a discretization of the cubic approximation based on the one proposed by Venturini et al. (2017). A set of different speeds S is defined that can be used by ships to travel between ports. The set of speeds correspond to reasonable and realistic speeds in a range around the design speed. Then, for each of the selected speeds $\delta \in S$ and ship i, a fuel consumption value ($\gamma_{i,\delta}$) measured in tons/nautical mile can be calculated based on the cubic approximation using the following equation (2).

$$\gamma_{i,\delta} = \frac{F(i,\delta)}{\delta} = \frac{\left(\frac{\delta}{\delta_i}\right)^3 \Gamma_i}{\delta} \tag{2}$$

3.2. Cost Structure

The MPBAP aims at optimizing the operational costs for both carriers and terminal operators. This Section defines the main costs involved in the problem context and describes to which stakeholder the costs are related. An overview of the main sources of cost and revenue for both carriers and terminal operators is shown in Figure 3.

As mentioned in Section 3.1, the main cost driver for a shipping line company is the fuel consumption which usually accounts for more than 50% of the carrier's total costs (Fagerholt and Psaraftis 2015). Another carrier related cost is the waiting time at anchorage (i.e., waiting to berth at port). As described by Chang et al. (2012), the waiting cost is not only the direct cost of being for longer time at a port, but also the resulting loss of potential income (i.e., opportunity cost). Regarding the service time at port, this is usually pre-established by a contract or when booking the port call. The cost may differ based on multiple factors such as the number for containers to be loaded and unloaded (i.e., quay crane moves) or the size of the ship and number of cranes required.
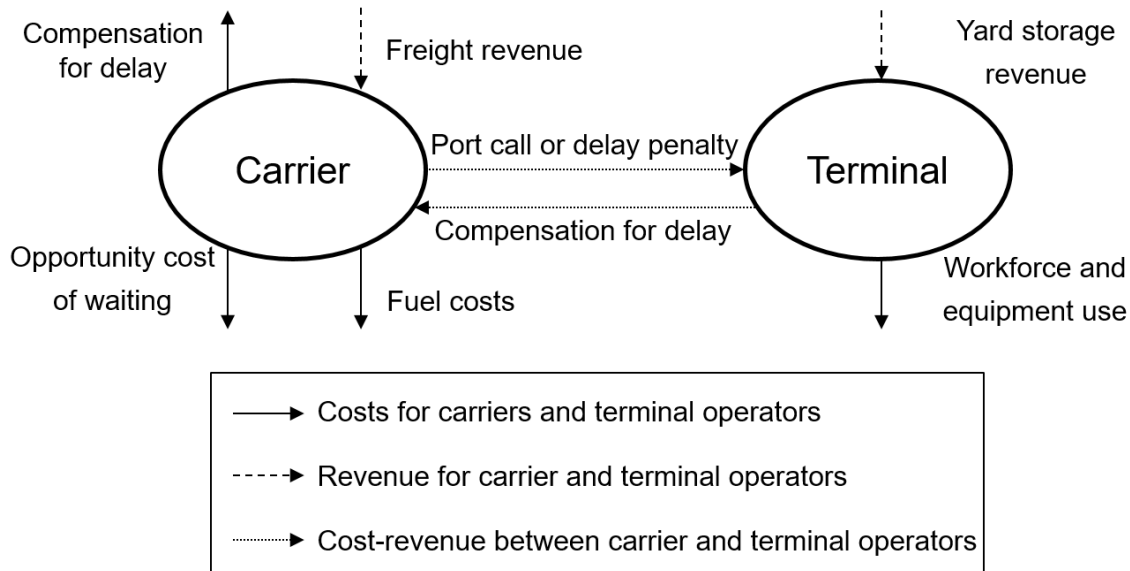
Figure 3   General overview of main costs and revenue for the shipping carriers and terminal operators.

In this case, the cost can be considered constant for the carrier regardless of the resulting quality of the terminal's planning. Finally, there are also delay costs associated to the carrier. Ending the service after the expected finish time at a port may result in additional payments to the shippers for the delay on the delivery of their cargo. In addition there may be other costs arising from delays. On one hand, if the delay at the terminal is caused by the ship arriving late, the carrier may be subject to a fine or delay penalty to the terminal. On the other hand, if the planned service time for a ship gets extended due to, for example, a breakdown of a quay crane or a poor berth allocation plan, the carrier affected by the delay may be entitled to a compensation from the terminal operator. It can be noticed, that these delay costs are paid from the carrier to the terminal operator or vice versa. This means that a cost for one party becomes a revenue for the other one. The main premise of this problem is that carriers and terminal operators jointly plan their operations and, therefore, the internal delay costs do not exist and can be excluded from the objective of the problem.

The main costs impacting the planning of the terminal in this problem are both the handling and delay costs. We identify the use of resources to be directly proportional to the number of workforce and quay crane usage hours. The fewer shifts needed to serve the vessel, the greater the profit is for the terminal. Therefore, both an increasing handling time by the vessels or an increased delay will require additional workforce. As mentioned before in this section, one of the premises of the MPBAP is that the planning decisions are agreed between the carrier and the terminal operator based on the overall best solution for all. This collaborative optimization removes the concept of delay between the participating players. However, we do consider a delay cost for the

terminal operators in the objective of the problem. We study the problem from a tactical point of view but assume that, for instance, workforce planning at the terminal is performed beforehand. In this scenario, the suggested optimal berth allocation plan for a given terminal may require more workforce than initially planned. This will directly translate in the use of additional resources to cover for the additional handling operations that can be computed as delay costs.

All in all, the MPBAP covers the costs depicted with a continuous line in Figure 3. Thus, the objective of the problem focuses on minimizing the fuel consumption and the costs related to waiting, handling and delay time.

3.3. Mixed-integer problem formulation. The Venturini et al. (2017) model

The solution method presented in this paper is based on a mixed integer problem (MIP) formulation from Venturini et al. (2017), which we now briefly present. We first list the notation used in the model:

Sets and parameters	
N	Set of ships
P	Set of ports
P_i	Set of ports to be visited by ship $i \in N$ sorted in visiting order
B_p	Set of berths at port $p \in P$
$V^{p,b}$	Set of vertices, $V^{p,b} = N \cup \{o(p,b), d(p,b)\}$, with $o(p,b) =$ origin node for arcs and $d(p,b) =$ destination node for arcs, both defined for every port $p \in P$ and berth $b \in B_p$
$A^{p,b}$	Set of arcs (i,j) with $i,j \in V^{p,b}, i \neq j$
S	Set of speeds
$Start_i^p$	Minimum starting time of activities for ship $i \in N$ at port $p \in P_i$
EFT_i^p	Expected finishing time of activities for ship $i \in N$ at port $p \in P_i$
$s^{p,b}$	Starting time of activities for berth $b \in B_p$ at port $p \in P$
$e^{p,b}$	Ending time of activities for berth $b \in B_p$ at port $p \in P$
$h_i^{p,b}$	Handling time of ship $i \in N$ at berth $b \in B_p$ at port $p \in P$
$d^{p,p'}$	Distance between pair of ports $p, p' \in P$
P_{iL}	The last port to be visited by ship $i \in N$ in the route
$\gamma_{i,\delta}$	Fuel consumption per unit of distance for ship $i \in N$ at speed $\delta \in S$
Δ_δ	Travelling time per unit of distance when travelling at speed $\delta \in S$
$M1^{p,b}$	Big-M value, $M1^{p,b} = e^{p,b}$
$M2_i^{p,b}$	Big-M value, $M2_i^{p,b} = e^{p,b} - h_i^{p,b}$
F_c	Fuel consumption cost in \$ per ton
H_c	Handling activities cost in \$ per hour
I_c	Idleness cost in \$ per hour
D_c	Delay cost in \$ per hour

Decision variables	
$y_{i,j}^{p,b} \in \mathbb{B}$	1 if ship j immediately succeeds ship i at berth $b \in B_p$ at port $p \in P$ where $(i,j) \in V^{p,b}$; 0 otherwise
$v_{i,\delta}^{p} \in \mathbb{B}$	1 if ship $i \in N$ sails from port p to some other port $p'(p, p' \in P_i := p \prec p')$ at speed $\delta \in S$; 0 otherwise
$T_{i}^{p,b} \in \mathbb{Z}^+$	Time at which ship $i \in N$ berths at berth $b \in B_p$ at port $p \in P_i$ (berthing time)
$T_{o(p,b)}^{p,b} \in \mathbb{Z}^+$	Time at which berth $b \in B_p$ at port $p \in P_i$ starts berthing ships (i.e., arrival time of the first ship to the berth)
$T_{d(p,b)}^{p,b} \in \mathbb{Z}^+$	Time at which berth $b \in B_p$ at port $p \in P_i$ finishes berthing ships (i.e., departure time of the last ship from the berth)
$T_{i}^{p} \in \mathbb{Z}^+$	Time at which port $p \in P_i$ opens activities for ship $i \in N$
$\Delta EFT_{i}^{p} \in \mathbb{Z}^+$	Difference between effective finishing time and EFT_i^p for ship $i \in N$ at port $p \in P_i$

The mathematical model is presented below:

$$\min \sum_{i \in N} \sum_{p,p' \in P_i \setminus \{P_{iL}\}:\{p \prec p'\}} I_c \left(\sum_{b \in B_{p'}} T_i^{p',b} - \sum_{b \in B_p} T_i^{p,b} + \sum_{b \in B_p} h_i^{p,b} \left(\sum_{j \in N \cup \{d(p,b)\}} y_{i,j}^{p,b} \right) - \sum_{\delta \in S} \Delta_\delta d^{p,p'} v_{i,\delta}^p \right)$$
$$+ \sum_{i \in N} \sum_{p \in P_i} \sum_{b \in B_p} H_c (h_i^{p,b} \sum_{j \in N \cup \{d(p,b)\}} y_{i,j}^{p,b}) + \sum_{i \in N} \sum_{p \in P_i} D_c \Delta EFT_i^p + \sum_{i \in N} \sum_{p,p' \in P_i \setminus \{P_{iL}\}:\{p \prec p'\}} \sum_{\delta \in S} F_c (\gamma_{i,\delta} d^{p,p'} v_{i,\delta}^p)$$
$$(3)$$

subject to:

$$\sum_{b \in B_p} \sum_{j \in N \cup \{d(p,b)\}} y_{i,j}^{p,b} = 1 \quad \forall i \in N, \forall p \in P_i \tag{4}$$

$$\sum_{j \in N \cup \{d(p,b)\}} y_{o(p,b),j}^{p,b} = 1 \quad \forall p \in P, \forall b \in B_p \tag{5}$$

$$\sum_{j \in N \cup \{o(p,b)\}} y_{j,d(p,b)}^{p,b} = 1 \quad \forall p \in P, \forall b \in B_p \tag{6}$$

$$\sum_{j \in N \cup \{d(p,b)\}} y_{i,j}^{p,b} - \sum_{j \in N \cup \{o(p,b)\}} y_{j,i}^{p,b} = 0 \quad \forall i \in N, \forall p \in P_i, \forall b \in B_p \tag{7}$$

$$T_i^{p,b} + h_i^{p,b} - \left(1 - y_{i,j}^{p,b}\right) M1^{p,b} \leqslant T_j^{p,b} \quad \forall (i,j) \in A^{p,b}, \forall p \in \{P_i \cap P_j\}, \forall b \in B_p \tag{8}$$

$$\sum_{b \in B_p} T_i^{p,b} + \sum_{b \in B_p} h_i^{p,b} \left(\sum_{j \in N \cup \{d(p,b)\}} y_{i,j}^{p,b} \right) + \sum_{\delta \in S} \Delta_\delta d^{p,p'} v_{i,\delta}^p \leqslant T_i^{p'} \quad \forall i \in N, \forall p, p' \in P_i \setminus \{P_{iL}\}: \{p \prec p'\} \tag{9}$$

$$T_i^p \geqslant Start_i^p \quad \forall i \in N, \forall p \in P_i \tag{10}$$

$$\sum_{b \in B_p} T_i^{p,b} + \sum_{b \in B_p} h_i^{p,b} \left(\sum_{j \in N \cup \{d(p,b)\}} y_{i,j}^{p,b} \right) - EFT_i^p \leqslant \Delta EFT_i^p \quad \forall i \in N, \forall p \in P_i \tag{11}$$

$$\sum_{b \in B_p} T_i^{p,b} \geqslant T_i^p \quad \forall i \in N, \forall p \in P_i \tag{12}$$

$$\left(\sum_{j \in N \cup \{d(p,b)\}} y_{i,j}^{p,b} + \sum_{j \in N \cup \{o(p,b))\}} y_{j,i}^{p,b} \right) M2_i^{p,b} \geqslant T_i^{p,b} \quad \forall i \in N, \forall p \in P_i, \forall b \in B_p \tag{13}$$

$$T_{o(p,b)}^{p,b} \geqslant s^{p,b} \quad \forall p \in P, \forall b \in B_p \tag{14}$$

$$T_{d(p,b)}^{p,b} \leqslant e^{p,b} \quad \forall p \in P, \forall b \in B_p \tag{15}$$

$$\sum_{\delta \in S} v_{i,\delta}^p = 1 \quad \forall i \in N, \forall p \in P_i \setminus \{P_{iL}\} \tag{16}$$

$$y_{i,j}^{p,b} \in \{0,1\} \quad \forall (i,j) \in A^{p,b}, \forall p \in P, \forall b \in B_p \tag{17}$$

$$v_{i,\delta}^p \in \{0,1\} \quad \forall i \in N, \forall p \in P_i, \forall \delta \in S \tag{18}$$

$$\Delta EFT_i^p, T_i^p \in \mathbb{Z}^+ \quad \forall i \in N, \forall p \in P_i \tag{19}$$

$$T_{o(p,b)}^{p,b}, T_{d(p,b)}^{p,b} \in \mathbb{Z}^+ \quad \forall p \in P, \forall b \in B_p \tag{20}$$

$$T_i^{p,b} \in \mathbb{Z}^+ \quad \forall i \in N, \forall p \in P_i, \forall b \in B_p \tag{21}$$

The objective function (3) minimizes the cost, both for the terminal operators and the liner shipping company. It consists of the four cost elements described in Section 3.2, namely, the cost of waiting at the port, the vessels' handling cost, the cost of delays, and the total cost of the fuel consumed when sailing between ports. The waiting time is computed as the positive difference between the berthing time and the arrival time whereas the delay is computed as the positive difference between the actual and expected berthing finish time. Constraints (4) ensure that each ship berths at only one berth at each port in its route. Constraints (5) and (6) denote that at each berth and each port, only one arc leaves the origin and one arrives at the destination respectively. The flow conservation for all arcs at each berth and each port is ensured by constraint (7). Constraints (8) guarantee that if ship j is berthing right after ship i, it waits until the handling is completed. The big-M values for these constraints can be tightened to the time when the berth closes ($e^{p,b}$). Constraints (9) ensure for each ship that the activities at the next port in the route do not commence before the ship arrives to the port. The left-hand side of the constraint computes the arrival time to the next port travelling at a chosen speed. The start of activities for each ship at each port must also start after the minimum allowed time ($Start_i^p$) as indicated in constraints (10). This also ensures that a ship cannot start berthing if it arrives too early. Both constraints (9) and (10) set a lower bound (LB) for the variable T_i^p. Constraints (11) compute and set the delay (ΔEFT_i^p) for each ship at each port. Constraints (12) ensure that the berthing time of each ship at each port starts after the activities for that ship are open at the port. The values of the berthing time variables for the not chosen berths are set to 0 by constraints (13). Constraints (14) and (15) ensure that all berthing periods occur within the time window of each berth. Constraints (16) ensure that exactly one speed is selected to travel between each pair of consecutive ports (leg) in the route. The domains for all the decision variables are defined in (17)-(21). We notice that a formulation where the time-based variables are defined as non-negative real numbers (i.e., $\mathbb{R}^+$) is

also valid. However, we maintain the integer property of the variables for a fair comparison with the presented methods and the formulation presented in Venturini et al. (2017).

This formulation contains a few modifications to the original model presented in Venturini et al. (2017) (referred to as *original* model). In the original model a set of additional variables for the arrival of a ship to a port is stated. These variables have been omitted in this formulation since the arrival time of a ship to the next port in the route is directly dependent on the departure time from the previous port and the sailing speed between ports. This calculation is given by the left-hand side of constraints (9), which then can be used to replace arrival time variables (e.g., in the objective function). The delay calculation constraints (11) use the berthing time ($T_i^{p,b}$) instead of the port opening time for the ship (T_i^p). The big-M value of constraints (8) is set to the closing time of the berth ($e^{p,b}$) instead of $e^{p,b} - min_{c \in (i,j)}\{Start_c^p\}$.

Venturini et al. (2017) enhance the original formulation by adding multiple sets of valid inequalities. These enhancements have also been implemented for the computational comparison. The reader is referred to the original publication for additional details.

3.4. Network flow formulation

The MPBAP can also be modeled as a network flow problem using a graph representation where each node represents a feasible berthing time at each port and berth and arcs enable the different combinations of berthing times along the route. This setup allows us to obtain a feasible voyage for a given ship by choosing a path along the ports in the graph. Figure 4 shows an illustrative example of such a path. It consists of three ports with either one or two berthing positions in each of them.

Let $G = (O, A)$ be a directed and acyclic graph formed by the sets of nodes O and arcs A. Additionaly, we define the subset of arcs $A^k \subseteq A$ which denote the arcs available for a given ship $k \in N$. Within the node set, we denote $o, d \in O$ as artificial source and sink nodes respectively. Let $\delta_k^+(u)$ be the set of nodes that can be reached by following a single outgoing arc $a \in A^k$ from node $u \in O$ for ship $k \in N$. Likewise, let $\delta_k^-(u)$ be the set of nodes that can be reached by following a single incoming arc $a \in A^k$ from node $u \in O$ for ship $k \in N$. Additionally, $\theta(u)$ denote the berthing time related to node $u \in O \backslash \{o, d\}$ and let $V(p, b) \subseteq O$ be the set of nodes corresponding to port $p \in P$ and berth $b \in B_p$. We use the notation $[x; y]$ to define an interval between x and y where y is included and $[x; y)$ where y is not. For each ship $n \in N$ port $p \in P$ berth $b \in B_p$ and operating time instant $t \in [s^{p,b}; e^{p,b})$, we define the set $C(n, p, b, t) \subseteq V(p, b)$ that denote the graph nodes for ship n whose berthing periods cover time t (i.e., nodes that are *in conflict* with any ship berthing at time t). This basically corresponds to the nodes of the previous $h_n^{p,b} - 1$ time instants and including the
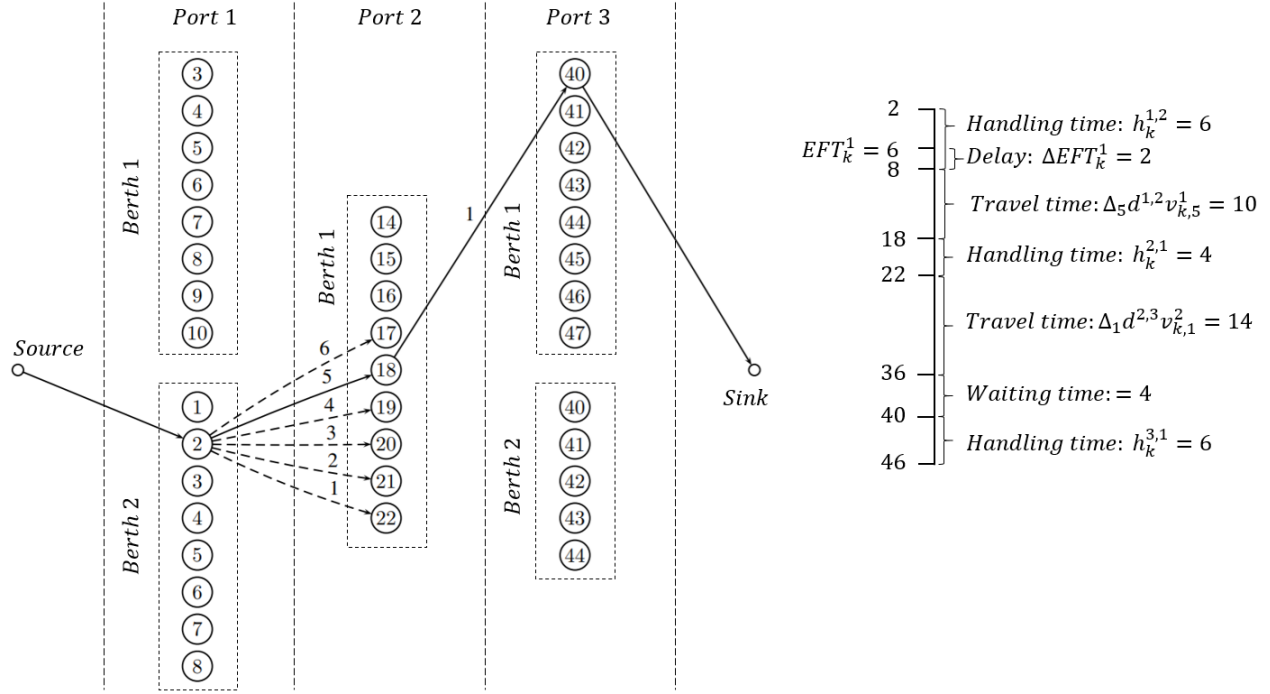
16

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

Figure 4 Example voyage for ship k and corresponding timeline. The number in the nodes indicate the berthing time and the number on the arcs denotes the speed level chosen. Alternative sailing options are denoted with dashed arcs. The rest of arcs in the graph are not displayed for simplicity.
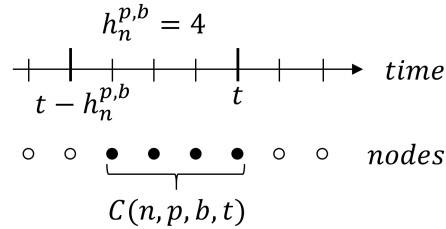

Figure 5 An example of the set $C(n, p, b, t)$ where the nodes depicted belong to $V(p, b)$ and refer to the time instant directly above. $h_n^{p,b}$ denotes the handling time for ship n.

node related to time t. An example is depicted in Figure 5 and the expression can be stated as follows:

$$C(n, p, b, t) := \left\{ v \in V(p, b) \middle| \theta(v) \in \left[\max\left(t - h_n^{p,b} + 1, s^{p,b} \right); \min\left(t, e^{p,b} \right) \right] \right\}$$

Finally, let $x_{i,j}^k$ be a binary variable deciding if arc $(i, j) \in A^k$ is selected for ship $k \in N$ and let $c_{i,j}$ be the weight associated to the same arc.

$$\min \sum_{k \in N} \sum_{(i,j) \in A^k} c_{i,j} x_{i,j}^k \tag{22}$$

$$\sum_{j \in \delta_k^+(o)} x_{o,j}^k = 1 \quad \forall k \in N \tag{23}$$

$$\sum_{i \in \delta_k^-(d)} x_{i,d}^k = 1 \quad \forall k \in N \tag{24}$$

$$\sum_{i \in \delta_k^-(j)} x_{i,j}^k - \sum_{i \in \delta_k^+(j)} x_{j,i}^k = 0 \quad \forall j \in O \backslash \{o,d\}, k \in N \tag{25}$$

$$\sum_{k \in N} \sum_{i \in C(k,p,b,t)} \sum_{j \in \delta_k^+(i)} x_{i,j}^k \leq 1 \quad \forall p \in P, b \in B_p, t \in [s^{p,b}; e^{p,b}) \tag{26}$$

$$x_{i,j}^k \in \{0,1\} \quad \forall (i,j) \in A, k \in N \tag{27}$$

The objective remains the same, and in this case the objective function (22) minimizes the cost of the selected arcs. Constraints (23) and (24) ensure that, for each ship, only one arc leaves from the source node and arrives to the sink node respectively. Constraints (25) enforce flow conservation ensuring that for each node, except the source and sink ones, there are as many incoming as outgoing arcs. Constraints (26) avoid overlapping of berthing periods in the same position by at most allowing one ship to be berthing at each time instant. Finally, constraints (27) define the binary property of the variable.

3.5. Generalized Set Partitioning Problem formulation

It is noted that all constraints of the network flow formulation (22)-(27) except constraint (26) are independent between ships. Exploiting the structure of the formulation, we can apply Dantzig-Wolfe decomposition (Dantzig and Wolfe (1960)) and transform it into a generalized set partitioning problem (GSPP) formulation where constraint (26) is handled in the master problem and each variable (i.e., column) refers to a whole feasible schedule of a ship along its route. According to Jans (2010), the pure binary nature of the variables of the network flow formulation allows us to impose binary conditions on the variables of the new master problem.

The set of all columns is comprised in Ω and the decision variable λ_j is set to 1 if column $j \in \Omega$ is chosen as part of the solution and 0 otherwise. We denote c_j as the cost related to column $j \in \Omega$. In order to replicate the objective of the MIP formulation, this cost consists of the idleness, handling cost, delay and bunker consumption cost of the ship denoted by the column. Let A_j^i be a parameter that is equal to 1 if column $j \in \Omega$ corresponds to ship $i \in N$ and 0 otherwise. Likewise, let $Q_j^{p,b,t}$ be a parameter that is equal to 1 if the ship of column $j \in \Omega$ is occupying berth $b \in B_p$ at time instant $t \in [s^{p,b}; e^{p,b})$ at port $p \in P$ and 0 otherwise.

$$\min \sum_{j \in \Omega} c_j \lambda_j \tag{28}$$

$$\sum_{j \in \Omega} A_j^i \lambda_j = 1 \quad \forall i \in N \tag{29}$$

$$\sum_{j \in \Omega} Q_j^{p,b,t} \lambda_j \leqslant 1 \quad \forall p \in P, b \in B_p, t \in [s^{p,b}; e^{p,b}) \tag{30}$$

$$\lambda_j \in \{0,1\} \quad \forall j \in \Omega \tag{31}$$

18

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

The objective function (28) minimizes the cost c_j of the columns. Constraints (29) ensure that one column is selected for each ship. Constraints (30) guarantee that, at each time instant, there is at most one ship berthing at each berth of a port. Finally, constraints (31) set the binary property of the decision variables.

4. Solution method

To solve (28)-(31), we propose a solution method based on a column generation procedure that, combined with branching, additional valid inequalities and symmetry breaking methods, results in a *branch-and-cut-and-price* algorithm.

4.1. Delayed Column generation

A common way of solving the GSPP formulation is by adding all the columns in advance. A successful example of this approach for the BAP can be found in Buhrkal et al. (2011). For the BAP instances presented, the amount of columns is manageable and can be easily pre-processed. However, in the MPBAP, the amount of columns increase exponentially with the multiple sailing speeds and ports for a ship. This makes the pre-processing intractable even for a few ports. Therefore, more dynamic strategies for handling the columns need to be explored. One efficient procedure is the so-called *delayed column generation*. This procedure relies on the premise that most of the variables will not be part of the optimal solution and, therefore, have a value of zero. Then, the focus is only on generating columns that have the potential to improve the objective value. This is done by relaxing and splitting the main problem into two, the master and subproblem. The restricted master problem (RMP) is the linear relaxation of the original formulation containing only a subset of the variables. The subproblem (or pricing problem) is used to identify the new variables. In our case, the relaxed version of the GSPP becomes the RMP and we define N independent subproblems, one per ship. The subproblem is defined as a shortest path problem in the network defined in Section 3.4 which can be solved in polynomial time. Since the graph is directed and acyclic (DAG), it can be solved by a DAG shortest path algorithm (see Cormen, Leiserson, and Rivest (1996) or Magnanti, Orlin, and Ahuja (1993)). The pricing problem aims at minimizing the reduced cost of a given path. At each iteration, after solving the RMP, the dual values of the RMP constraints are used to solve the pricing problems. We denote α_k to the dual variable for ship $k \in N$ associated to constraint (29). Likewise, we denote $\mu_{p,b,t}$ to the dual variable for port $p \in P$, berth $b \in B_p$ and time $t \in [s^{p,b}; e^{p,b})$ associated to constraint (30). Let $\bar{\alpha}_k, \bar{\mu}_{p,b,t}$ be the dual solution values for the RMP and let Λ_j be a sequence of *(port,berth,time)* elements. Each of these elements refers to the port, berth and time of a graph node visited by column $j \in \Omega$. The reduced cost $\hat{c}_j$ for a specific path j for ship $k \in N$ is computed as follows:

$$\hat{c}_j = c_j - \Big(\sum_{(p,b,t) \in \Lambda_j} \sum_{t' \in [t; t+h_k^{p,b})} \bar{\mu}_{p,b,t'} \Big) - \bar{\alpha}_k$$

Finally, for each pricing problem, we add the path with the lowest reduced cost to the RMP only if $\hat{c}_j$ is negative (i.e, $\hat{c}_j < 0$).

In fact, when the pricing problem is a pure shortest path problem, the LP bound arising from solving the GSPP with column generation and solving the LP relaxation of the network flow problem is the same. This indicates that the Dantzig-Wolfe decomposition does not provide any gain bound-wise. On the other hand, in cases of very dense networks with significantly more arcs than nodes as in our case, solving the GSPP with column generation is expected to be faster (e.g., see Brouer, Pisinger, and Spoorendonk (2011)).

4.2. Branching

Since the decision variables of the RMP are linear, the solution at the root node is often fractional and branching methods are required in order to achieve integrality. A major aspect of the branching procedure is selecting a branching candidate, whose branch children improve the lower bound the most. The most common branching methods consider branching on a specific node or arc from the graph. These strategies can be effective in some cases but do not necessarily apply to our problem. For instance, when branching on a graph node, one child will enforce the graph node to be used in the subsequent branch-and-bound (B&B) tree while the other child will forbid it. Considering the large amount of nodes for most instances in this problem, we can clearly see that the effect can be significant for the first child but rather minimal for the second. This often results in a highly unbalanced B&B tree to explore. In this study, we present a different branching strategy for the problem at hand that aims to be more effective than branching on a single graph node.

The proposed branching strategy states that, given a fractional solution, we compute, for each ship n and port p, the average berthing time t and the variance of these times among all solution columns. As an example, consider a fractional solution containing two columns for ship 1. At port 1, these columns correspond to ship 1 berthing at time 4 and 6 respectively. Then, for ship 1 and port 1, the average berthing time is 5 whereas the sample standard deviation is $\sqrt{2}$. We define this average berthing time and variance as a *candidate* which results in a total of $|N| \cdot |P|$ candidates. We then select the candidate whose variance of berthing times is higher. The procedure is described in Algorithm 1. Each of the child branches will enforce ship n to berth before or after time t respectively at port p. It should be noticed that a fractional solution where ships berth at the same time but at different berthing positions can exist. In this case, we can obtain candidates with no variance resulting in an impractical branching. If that happens, the criterion is changed to branching on berthing positions instead of on berthing times, following the same procedure. In practice, this scenario is highly unlikely to happen and we have not experienced it in any experiments hitherto. As a result, the proposed strategy opts for branching on a set of graph nodes instead of on a single one.

20

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

Finally, the B&B tree is explored following a *best first* policy. This policy prioritizes the queue of unexplored nodes according to their bound. Thus, the next node to be explored is always the one with the *best* (i.e., lowest) lower bound.

Algorithm 1: Branching candidate selection

Data: *sol*: current solution.

Result: $Cand^*$: the candidate to branch on.

1 begin
2 $[\lambda] \leftarrow sol$ // classify solution columns (λ) by ship
3 $Cand^* = \emptyset$ // initialize best candidate
4 $\sigma^* = 0$ // initialize standard deviation of candidate's berthing times
5   **for** *ships and ports* **do**
6 $[times] \leftarrow \lambda(ship, port)$ // set of solution berthing times at *port* for *ship*
7 $time \leftarrow avg([times])$ // get average of berthing times
8   **if** $\sigma([times]) > \sigma^*$ **then**   // compare the standard deviation with the current best
9 $\sigma^* \leftarrow \sigma([times])$
10 $Cand^* \leftarrow time, port, ship$ // update best candidate so far

---

4.3. Valid inequalities

In order to improve the lower bound, we propose a set of valid inequalities that can be added to the problem by separation.

Figure 6 shows a small LP solution to a trivial problem with two ships (i.e., continuous and dashed lines), one port and one berth where an example of a violated valid inequality can be found. We define u, v as the two nodes corresponding to ship A (berthing at times 1 and 3) and let w be the node of ship B berthing at time 2. We observe that the arc from node w is in conflict with the arcs from both nodes u and v due to overlapping berthing periods. In other words, the berthing period of ship B at node w covers, at least partially, both berthing periods of ship A at nodes u and v. The arcs from u, v are also in conflict with each other as they belong to the same ship. As a result, we notice that, at most, one outgoing arc can be chosen out of the ones from these three nodes. Since the solution values of the outgoing arcs sum to 1.5, this valid inequality would cut the example LP solution. We aim at generalizing the definition of such a valid inequality and introduce the following proposition:
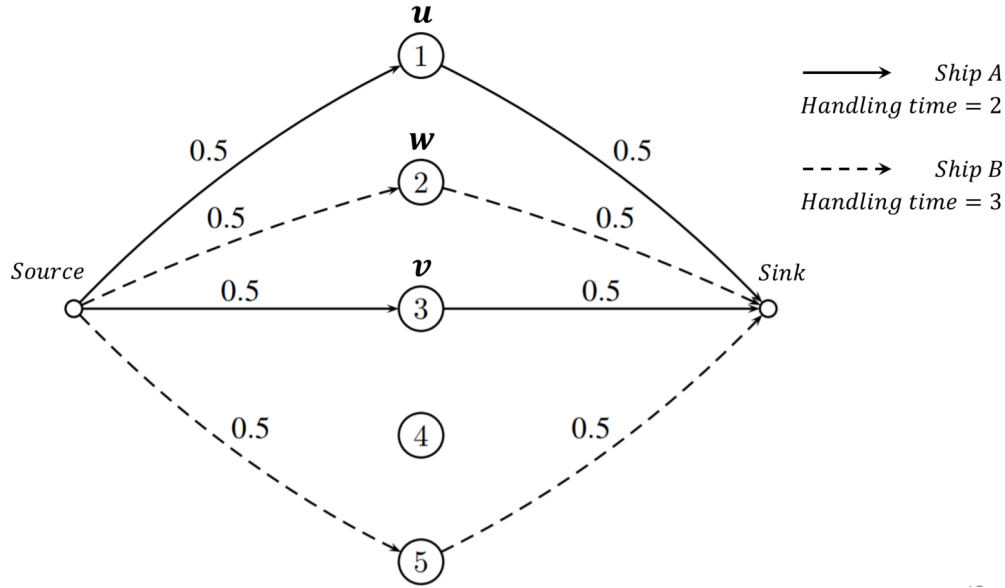
Figure 6   Example LP solution of a problem with two ships, one port and one berth. The nodes represent berthing
times and the numbers on the arcs denote the solution value of the arc variable $x_{i,j}^k$.

PROPOSITION 1. *Given two time instants $t_1, t_2 \in [s^{p,b}; e^{p,b})$ where $t_1 < t_2$ and a port $p \in P$, berth $b \in B_p$ and ship $n \in N$, the following is a valid inequality:*

$$\sum_{u \in \bigcup_{t \in [t_1;t_2]} C(n,p,b,t)} \sum_{w \in \delta_n^+(u)} x_{u,w}^n + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,b,t_1) \cap C(m,p,b,t_2)} \sum_{w \in \delta_m^+(u)} x_{u,w}^m \leqslant 1$$

Proof.   The set $C(m,p,b,t)$ used in constraint (26) defines the set of nodes for ship m that are in conflict with time t (see Section 3.4). Based on this definition, the intersection set $C(m,p,b,t_1) \cap C(m,p,b,t_2)$ directly defines the set of nodes for ship m that are in conflict with both time instants t_1 and t_2. Constraint (26) indicates that at most one arc can be chosen out of the nodes from the sets $C(m,p,b,t)$ of all ships $m \in N$ and, therefore, the same applies to the intersection set $C(m,p,b,t_1) \cap C(m,p,b,t_2)$. By considering the intersection set $C(m,p,b,t_1) \cap C(m,p,b,t_2)$ for all ships except one $m \in N \setminus \{n\}$, the berthing period for ship n is only required to be in conflict with either t_1 or t_2 and can be defined as the union of $C(n,p,b,t_1) \cup C(n,p,b,t_2)$. Considering these node sets, we can define the following valid inequality:

$$\sum_{u \in C(n,p,b,t_1) \cup C(n,p,b,t_2)} \sum_{w \in \delta_n^+(u)} x_{u,w}^n + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,b,t_1) \cap C(m,p,b,t_2)} \sum_{w \in \delta_m^+(u)} x_{u,w}^m \leqslant 1$$
$$\forall p \in P, b \in B_p, n \in N, t_1, t_2 \in [s^{p,b}; e^{p,b}), t_1 < t_2$$

Based on the assumption that a berthing period cannot be discontinued, the intersection set $C(m,p,b,t_1) \cap C(m,p,b,t_2)$ for any ship is not only in conflict with times t_1 and t_2 but with all the time instants in the period $[t_1; t_2]$. Therefore the interval for ship n can be expanded to the union
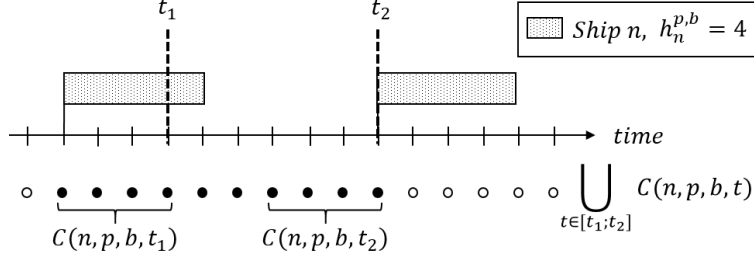
Figure 7   In a valid inequality for ship n, port p, berth b and time instants t_1, t_2, the filled nodes indicate the interval for ship n with handling time $h_n^{p,b}$. The rectangles indicate the berthing period of ship n at the earliest and latest possible berthing times in the interval.

of $C(n, p, b, t)$ sets for all time instants $t \in [t_1; t_2]$. An example of this set is shown in Figure 7 and the resulting valid inequality can be formulated as follows:

$$\sum_{u \in \bigcup_{t \in [t_1;t_2]} C(n,p,b,t)} \sum_{w \in \delta_n^+(u)} x_{u,w}^n + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,b,t_1) \cap C(m,p,b,t_2)} \sum_{w \in \delta_m^+(u)} x_{u,w}^m \leqslant 1$$

$$\forall p \in P, b \in B_p, n \in N, t_1, t_2 \in [s^{p,b}; e^{p,b}), t_1 < t_2 \quad (32)$$

☐

Returning to the example in Figure 6, the mentioned cut would be included in the proposed valid inequality (32) for $n = A$, $t_1 = 2$ and $t_2 = 4$ where node w would correspond to a node from the intersection sets $C(B, p, b, 2) \cap C(B, p, b, 4)$ and nodes u, v for ship A would correspond to berthing times covering t_1 and t_2 respectively and therefore belonging to the set $\bigcup_{t \in [2;4]} C(A, p, b, t)$.

We note that the inequality only is interesting when $C(m, p, b, t_1) \cap C(m, p, b, t_2) \neq \emptyset$. The size of the intersection set is dependent on the time instants t_1, t_2 used and we observe that this size increases when the t_1, t_2 are closer together in time.

These valid inequalities (32) are added by separation after the column generation procedure concludes. Exploring the entire set of valid inequalities can be computationally intensive. Therefore, only valid inequalities based on berthing times from the LP solution are checked since the arcs from the related nodes are guaranteed to contain non-zero values and the resulting inequalities have a higher probability of being violated by the LP solution. Given an LP solution, let t_1^* and t_2^* be two berthing times for ship n at berth b of port p where $t_1^* \leq t_2^*$. Let t_3^* be a berthing time for another ship m at the same berth b of port p whose berthing period is both in conflict with t_1^* and t_2^* for ship n. The conditions that t_3^* needs to satisfy to be in conflict with t_1^* and t_2^* are given by the following inequalities:

$$t_1^* + h_n^{p,b} > t_3^*$$
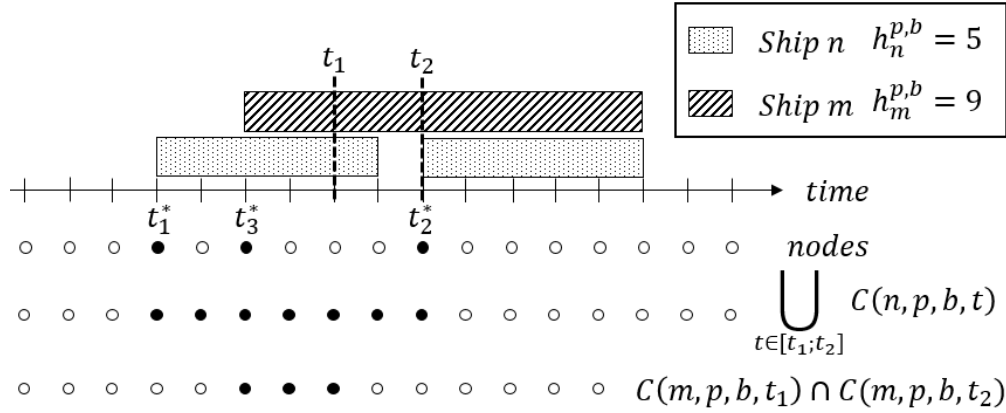
$$t_2^* < t_3^* + h_m^{p,b}$$

Figure 8 Example of times t_1, t_2 definition based on solution times t_1^*, t_2^* for ship n and t_3^* for ship m. The bottom last two rows of filled nodes define the node interval for ship n and m respectively.

Based on these times, we can calculate time instants t_1, t_2 for a valid inequality that includes t_1^*, t_2^* for ship n and t_3^* for ship m as follows:

$$t_1 = t_1^* + h_n^{p,b} - 1, \quad t_2 = t_2^*$$

An example of this calculation is shown in Figure 8. It can be noticed that the interval for ship n starts at time t_1^* and ends at time t_2^*. If we add such a violated cut to the RMP, we risk finding a very similar solution in the next iteration where columns are shifted, for example, one time instant before t_1^* or after t_2^*. In order to avoid that, we aim at defining time instants t_1, t_2, so that the resulting intervals do not only cover solution nodes but also a number of neighboring nodes related to time instants immediately before and after the solution time. We aim at expanding the interval between t_1^* and t_2^* as well as the one around t_3^*. Based on the inequalities aforementioned to ensure that t_1^*, t_2^* and t_3^* relate to conflicting periods, we introduce the slack variables Δ^X and Δ^Y that would indicate how much we can modify the node intervals.

$$t_1^* + h_n^{p,b} > t_3^* + \Delta^X$$

$$t_2^* + \Delta^Y < t_3^* + h_m^{p,b}$$

Both slack values are distributed equally between both intervals, which leads us to the following calculation of t_1 and t_2:

$$t_1 = t_1^* - \frac{\Delta^X}{2} + h_n^{p,b} - 1, \quad t_2 = t_2^* + \frac{\Delta^Y}{2}$$

Due to the discretization of the time horizon, if $\frac{\Delta^X}{2}$ or $\frac{\Delta^Y}{2}$ is fractional, they are rounded-up in the calculation of t_1 and t_2. Also, in the case that there is limited room for expansion in one of the intervals (e.g., operational time windows), the remaining slack is added to the other interval.
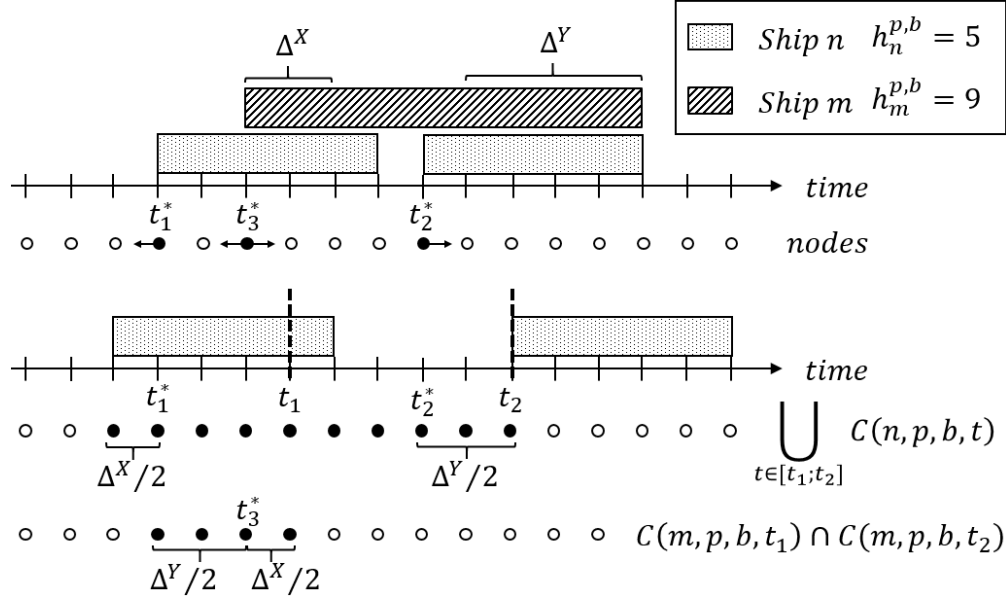
Figure 9   Example of times t_1, t_2 selection based on solution times t_1^*, t_2^* for ship n and t_3^* for ship m. **The upper illustration depicts the LP solution for the three berthing times selected and the available slack and direction of expansion for the desired node intervals. The lower illustration depicts the resulting times t_1, t_2 for the valid inequality and the respective node intervals for ships n and m.**

Figure 9 shows an example of the calculation of times t_1 and t_2 based on solution times t_1^*, t_2^* and t_3^* and slack variables Δ^X, Δ^Y.

In order to ensure $t_1 < t_2$, by substituting the above expressions, the criterion that t_1^*, t_2^*, t_3^* need to fulfill in order to result in a valid inequality can be defined as follows:

$$t_1^* - \frac{\Delta^X}{2} + h_n^{p,b} - 1 < t_2^* + \frac{\Delta^Y}{2}$$

Not satisfying this inequality leads to a cut that, at best, is equal to constraint (30) which is already present in the RMP.

The entire cut separation process is summarized in Algorithm 2. The procedure requires the RMP model and an LP solution as input. From the solution, both the λ^* solution values and the berthing times of the solution columns are extracted and classified by ship, port and berthing position. The cuts are checked by enumerating combinations of solution times t_1^*, t_2^* and t_3^*. Only solution times whose berthing periods are in conflict are considered. This is the case if the berthing period of ship m at time t_3^* overlaps both berthing periods of ship n at times t_1^* and t_2^* ($inConflict(t_1^*, t_2^*, t_3^*)$ in Algorithm 2). Then, the solution times are used to compute time instants t_1, t_2 distributing the slack available as aforementioned in this Section ($t_1, t_2 \leftarrow calcInterval(t_1^*, t_2^*, t_3^*)$ in Algorithm

2). To check and add the violated cuts to the RMP, equation (32) needs to be translated to the problem variables. The $x_{i,j}^k$ variables can be defined using λ_p variables as follows:

$$x_{i,j}^k = \sum_{p \in \Omega} q_{i,j,p}^k \lambda_p$$

where parameter $q_{i,j,p}^k$ is 1 if graph arc $(i,j) \in A^k$ for ship k is used by column $p \in \Omega$ and 0 otherwise. Applying this equality to equation (32), we obtain the following version of the equation:

$$\sum_{u \in \bigcup_{t \in [t_1;t_2]} C(n,p,b,t)} \sum_{w \in \delta_n^+(u)} \sum_{j \in \Omega} q_{u,w,j}^n \lambda_j + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,b,t_1) \cap C(m,p,b,t_2)} \sum_{w \in \delta_m^+(u)} \sum_{j \in \Omega} q_{u,w,j}^m \lambda_j \leq 1 \quad (33)$$

For each cut inspected, the left-hand side of constraint (33) is computed and the valid inequality is added to the RMP if it is violated.

Algorithm 2: Cut separation

Data: sol, RMP: current solution and model.

Result: RMP: updated model with separated cuts.

1 begin
2 $times[p,b,n] \leftarrow sol$ // classify solution times by port p, berth b and ship n
3 $[\lambda^*] \leftarrow sol$ // obtain solution values for columns
4   **for** $p \in P, b \in B_p, n \in N$ **do**   // cuts are based on a specific berth, port and ship
5   **for** $t_1^*, t_2^* \in times[p,b,n]$ **do**   // loop over pairs of solution times for ship n
6   **for** $m \in N, m \neq n$ **do**
7   **for** $t_3^* \in times[p,b,m]$ **do**   // select a third time from a different ship
8   **if** $inConflict(t_1^*, t_2^*, t_3^*)$ **then**   // check if berthing periods are in conflict
9 $t_1, t_2 \leftarrow \text{calcInterval}(t_1^*, t_2^*, t_3^*)$ // compute t_1, t_2 for the valid inequality
10 $violatedCut \leftarrow \text{checkCut}(t_1, t_2, n, p, b, [\lambda^*])$ // add violated cut to the RMP

These valid inequalities are relatively easy to handle in the reduced cost computation. For each valid inequality, its corresponding dual value needs to be subtracted in each of the nodes considered for each ship in the constraint. As an example, given a valid inequality for times t_1, t_2 where $t_1 < t_2$, port p, berth b and ship n, its dual value needs to be subtracted in nodes $\bigcup_{t \in [t_1;t_2]} C(n,p,b,t)$ for ship n and in nodes $C(m,p,b,t_1) \cap C(m,p,b,t_2)$ for ship m where $m \neq n$. A more mathematical definition of the updated reduced cost computation is given in Appendix A.

4.4. Symmetry breaking

In some instances, at each port, some of the berthing positions are identical in terms of their availability time window and the handling times for all ships. Identical berths may lead to many equivalent solutions, which may increase the solving time of the model. Therefore, we propose adapting the model so it deals with berth types instead of individual berths in a similar procedure as the one stated in Buhrkal et al. (2011). Let K_p be the set of berth types for port $p \in P$ and β^k be the number of berthing positions of type $k \in K$ in the problem. For each berth type $k \in K_p$ at port $p \in P$, $s^{p,k}$ and $e^{p,k}$ denote its opening and closing time respectively and the parameter $Q_j^{p,k,t}$ is 1 if the ship from column $j \in \Omega$ occupies berth type k at time instant $t \in [s^{p,k}; e^{p,k})$ at port p and 0 otherwise. We can therefore update the set of constraints (30) as follows:

$$\sum_{j \in \Omega} Q_j^{p,k,t} \lambda_j \leqslant \beta^k \quad \forall p \in P, k \in K_p, t \in [s^{p,k}; e^{p,k}) \tag{34}$$

This adaptation has an equivalent impact in constraints (26) from the network formulation where the right-hand side is also replaced by β^k. The valid inequality (33) from Proposition 1 can be updated similarly and it is described in Proposition 2 that can be found in Appendix B. The resulting valid inequality is formulated as follows:

$$\sum_{u \in \bigcup_{t \in [t_1; t_2]} C(n,p,k,t)} \sum_{w \in \delta_n^+(u)} \sum_{j \in \Omega} q_{u,w,j}^n \lambda_j + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,k,t_1) \cap C(m,p,k,t_2)} \sum_{w \in \delta_m^+(u)} \sum_{j \in \Omega} q_{u,w,j}^m \lambda_j \leq \beta^k$$
$$\forall p \in P, k \in K_p, n \in N, t_1, t_2 \in [s^{p,k}; e^{p,k}), t_1 < t_2 \tag{35}$$

The reduced cost computation is also slightly modified where the dual variable $\mu_{p,k,t}$ of the modified constraint now is based on berth type $k \in K_p$ instead of berth $b \in B_p$.

We expect to see an improvement in the computational time as soon as there are two identical berths at a port. Likewise, we expect to see larger symmetry for the instances containing more berthing positions per port.

5. Cooperative game theory

The MPBAP is based on a strong collaboration between carriers and port operators and some of them, especially carriers, may be reticent to take part in such a collaboration scheme. In order to convince them that this form of collaboration is beneficial for all of them, we define a cooperative game. The aim is to show that all stakeholders (i.e., carriers and terminals) can potentially benefit from a collaboration by distributing the overall costs efficiently. Our cooperative game is formed by a set of players $\mathcal{P} = \{1, ..., p\}$, which in this case corresponds to both the carriers owning the ships and the terminal operators of the ports visited by the ships. The *characteristic function* $\vartheta(S)$ measures the impact of a coalition of players $\mathcal{S} \subseteq \mathcal{P}$, which in this case is measured by the

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

27

operational costs. The coalition formed by all players is known as the *grand coalition*. It is normally assumed that the characteristic function satisfies:

$$\vartheta(\emptyset) = 0 \tag{36}$$

$$\vartheta(\mathcal{S} \cup \mathcal{T}) \leq \vartheta(\mathcal{S}) + \vartheta(\mathcal{T}) \quad \forall \mathcal{S}, \mathcal{T} \subseteq \mathcal{P}, \quad \mathcal{S} \cap \mathcal{T} = \emptyset \tag{37}$$

Equation (36) states that an empty coalition has a cost of zero, while equation (37), known as *subadditivity*, indicates that the costs of two separate coalitions $\mathcal{S}, \mathcal{T} \subseteq \mathcal{P}$ cannot be lower than when acting together. A solution to a cooperative game (i.e., *imputation*) can be defined as $f = \{f_1, ..., f_p\}$ where f_i corresponds to the cost allocation of player i in coalition $\mathcal{P}$. An imputation should satisfy the following conditions:

$$f_i \leq \vartheta(\{i\}) \quad \forall i \in \mathcal{P} \tag{38}$$

$$\sum_{i \in \mathcal{P}} f_i = \vartheta(\mathcal{P}) \tag{39}$$

The first condition is based on individual rationality and defines that the cost allocation for a player when being part of the grand coalition cannot be worse than the player's standalone cost. The second condition is based on group rationality and states that all the savings arising from a grand coalition are shared. This is the equivalent of saying that the sum of cost allocations needs to be equal to the total cost of the grand coalition and a solution fulfilling this condition is said to be *efficient*. Furthermore, we consider a solution to be *stable*, if, for every coalition $\mathcal{S} \subseteq \mathcal{P}$, the sum of allocated cost of the players of the coalition is not higher than the cost of the coalition $\sum_{k \in \mathcal{S}} f_k \leq \vartheta(\mathcal{S})$. We define the *core* as the set of solutions that are both *efficient* and *stable*. We see the core solutions as the most attractive and fair for all players. Note, however, that the core may be empty in some cases. This means that a cost allocation that satisfies both the efficiency and stability properties does not exist. In other words, it means that a subset of the players in the grand coalition could do better by themselves (i.e., by forming a sub-coalition). If the core is empty, the grand coalition is unstable and there is a risk that it breaks apart. In practice, the grand coalition may stay together despite a non-core solution. For instance, it may be that a subset of players are not aware of the higher benefits of a specific sub-coalition or that players choose to stay in the coalition to reap more long-term benefits given future expectations. Next, we describe the two allocation methods we have used in this study.

5.1. Shapley value
The Shapley value (Shapley 1953), refers to the weighted average of each player's marginal contribution to each of the potential coalitions. Let $\Theta^i(\mathcal{S})$ be the marginal contribution of player i to

coalition $\mathcal{S}$, which is seen as the difference between the cost of the coalition including player i and the coalition without the player:

$$\Theta^i(\mathcal{S}) = \vartheta(\mathcal{S} \cup \{i\}) - \vartheta(\mathcal{S}) \tag{40}$$

Then, the cost allocated to participant i is computed by the following expression:

$$f_i = \sum_{\mathcal{S} \subseteq \mathcal{P} \setminus \{i\}} \frac{|\mathcal{S}|! |\mathcal{P} \setminus (\mathcal{S} \cup \{i\})|!}{|\mathcal{P}|!} \Theta^i(\mathcal{S}) \tag{41}$$

where $|\cdot|$ refers to the number of players in the given coalition. Once the characteristic function $\vartheta(\mathcal{S})$ is calculated for all possible coalitions $\mathcal{S}$, it is a simple method to compute as it only requires applying a formula. The Shapley value does not only provide *efficient* solutions, it also contains other valuable properties. The solutions are *symmetric* meaning that if two players contribute equally to the coalitions, they achieve the same savings. *Anonimity* is also ensured, which states that the order or labelling of players does not have an impact on the assignment of savings. This property ensures a unique solution which avoids players to regret their choices and prevents additional negotiation processes. On the other hand, the Shapley value does not ensure the *stability* property, meaning that the solution is not guaranteed to be part of the core.

5.2. Equal profit method (EPM)

The goal of the equal profit method (Frisk et al. 2010) is to find the solution in the core that minimizes the maximal difference in relative savings between pairs of players. The relative saving of player i is computed as $\frac{\vartheta(\{i\}) - f_i}{\vartheta(\{i\})}$. The method is formulated as the following linear programming model:

$$\min z \tag{42}$$

$$z \geq \frac{f_i}{\vartheta(\{i\})} - \frac{f_j}{\vartheta(\{j\})} \quad \forall i, j \in \mathcal{P} \tag{43}$$

$$\sum_{i \in \mathcal{P}} f_i = \vartheta(\mathcal{P}) \tag{44}$$

$$\sum_{i \in \mathcal{S}} f_i \leq \vartheta(\mathcal{S}) \quad \forall \mathcal{S} \subseteq \mathcal{P} \tag{45}$$

$$f_i \geq 0 \quad \forall i \in \mathcal{P} \tag{46}$$

Constraints (43) calculate the difference in relative savings between each pair of players and restricts z to the largest of those differences. Note that constraints (44) and (45) are the ones denoting the stability and efficiency properties which means that the EPM method only allows solutions lying in the core.

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

29

6. Computational results

This section is divided in two. First, the performance of the proposed method is compared to a commercial solver on the set of instances from Venturini et al. (2017) and an additional generated set of harder instances. The second part covers the results of the cost allocation methods for the cooperative game.

6.1. Instance results

Different versions of the algorithm have been tested varying the size of the B&B tree where valid inequalities can be added. We consider (i) a pure *branch-and-price* where cut separation is not performed at all, (ii) a partial *branch-and-cut-and-price* where we only allow valid inequalities to be added in the root node, and (iii) a pure *branch-and-cut-and-price* where cuts can be added in all the explored nodes. The RMP model solved is comprised by equations (28),(29),(34), the linear relaxation of (31) and valid inequalities (35) that are added by separation. The algorithm includes a running time-limit and, if it is reached and a gap between the lower and upper bounds still exists, the GSPP formulation problem is solved with all the generated columns in the B&B tree. This helps tightening the upper bound but requires the integer problem to be solvable in reasonable time. The running time for solving the GSPP is set to 10% of the algorithm running time. Two algorithm time limits of 5 minutes and 3 hours have been tested with an additional (if required) 30 seconds and 18 minutes respectively for solving the GSPP. The model has been entirely written in *Julia* language (Bezanson et al. 2017), modelled using *JuMP* (Dunning, Huchette, and Lubin 2017) and using *CPLEX v. 12.9* as the solver, allowing 4 threads. It has been tested in an 2.20 GHz Intel Xeon Processor 2650v4 using 4 cores with 32 GB of memory per core. The MIP formulation from Venturini et al. (2017) has been run in the same machine and solved with the same solver for a fair comparison. The results are summarized in Tables 1, 2, 3 and 4, that contain the performance comparison on the benchmark instances from Venturini et al. (2017) and the generated set of harder instances with both algorithm time limits. An instance is represented indicating the number of ships N, the number of berthing positions per port B, the number of ports P and if the time windows TW are tight T or loose L. As indicated in Venturini et al. (2017) a loose time window is approximately 3 times longer than a tight one. In each instance, all ports have the same amount of berthing positions and all the ships follow the same route and have the same speed profiles but both the MIP and GSPP formulations can account for different amount of berthing positions per port, different ship routes and different ship types. The set S is discretized in 11 speed levels, covering the range 14-19 knots. A very low sulphur fuel oil (VLSFO) is used by the ships which is in accordance with the increasing need of ships to reduce their sulphur emissions. Its price (F_c) is computed as the average global price during the first quarter of 2021 corresponding to 500 \$/ton

(Ship & Bunker 2021). Regarding the cost of the different operational aspects at port, the current literature does not provide a consensus on the costs of waiting, handling and delay time. Moreover, this may fluctuate significantly between ports and in many cases they are not made available to the public due to contractual agreements. Meisel and Bierwirth (2009) proposes a delay cost of 1000-3000 $/hour depending on the ship size and a service cost per quay crane hour of 100 $. They also consider a speeding-up cost to berth at an earlier time of 1000-3000 $/hour which can resemble the waiting time cost considered in this study. Venturini et al. (2017) set the terminal handling cost weight to 200 $/hour and charge an additional 300 $/hour when there is a delay. They set the cost of waiting one hour at anchorage to 200 $/hour. For the sake of a fair comparison, we use the same costs as Venturini et al. (2017) which correspond to $H_c = 200$, $D_c = 300$ and $I_c = 200$. LB denotes the best lower bound found whereas Z indicates the best integer solution (i.e., upper bound). The optimality gap is stated under the column Gap and it is calculated using the optimal solution, or in the case that this is unknown, the best known solution. The computational time in seconds is given under column T.

Table 1 Computational results on instances from Venturini et al. (2017) with a total time limit of 5 minutes and 30 seconds. The MIP formulation is compared to the variants of the presented branch-and-cut-and-price method. "-" means that no integer solution has been found within the time limit. "*" means the time limit has been reached. The best running time is highlighted in bold for instances solved to optimality and the best optimality gap for the rest of instances.

Instance	MIP formulation				Branch & Price				Branch & Cut (root node) & Price				Branch & Cut & Price			
N-B-P-TW	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)
4-3-3-L	296,600	296,600	0.00	**0.1**	296,600	296,600	0.00	0.5	296,600	296,600	0.00	0.2	296,600	296,600	0.00	0.2
5-3-3-L	394,300	394,300	0.00	**0.4**	394,300	394,300	0.00	3.2	394,300	394,300	0.00	7.2	394,300	394,300	0.00	7.2
6-3-3-L	421,720	421,720	0.00	2.3	421,720	421,720	0.00	0.8	421,720	421,720	0.00	**0.5**	421,720	421,720	0.00	**0.5**
6-3-4-L	647,480	647,480	0.00	89.2	647,480	647,480	0.00	**39.0**	647,480	647,480	0.00	111.7	647,480	647,480	0.00	103.7
10-4-4-L	1,014,437	1,060,900	3.80	*	1,053,030	1,054,700	0.14	*	1,053,092	1,055,300	0.13	*	1,053,295	1,055,000	**0.11**	*
10-4-3-L	689,858	700,000	1.18	*	698,100	698,100	0.00	193.6	698,100	698,100	0.00	**72.7**	698,100	698,100	0.00	115.6
4-4-4-L	405,120	405,120	0.00	**0.3**	405,120	405,120	0.00	0.6	405,120	405,120	0.00	0.6	405,120	405,120	0.00	0.6
5-4-4-L	500,600	500,600	0.00	**0.4**	500,600	500,600	0.00	0.9	500,600	500,600	0.00	0.8	500,600	500,600	0.00	0.8
6-4-4-L	599,980	599,980	0.00	**1.2**	599,980	599,980	0.00	7.6	599,980	599,980	0.00	5.1	599,980	599,980	0.00	5.3
12-5-3-L	811,139	840,640	2.32	*	830,440	830,440	0.00	136.0	830,440	830,440	0.00	**112.1**	830,440	830,440	0.00	120.4
10-6-3-L	680,600	680,600	0.00	219.4	680,600	680,600	0.00	9.7	680,600	680,600	0.00	**5.1**	680,600	680,600	0.00	5.8
11-6-3-L	740,430	749,620	0.78	*	746,220	746,220	0.00	22.5	746,220	746,220	0.00	**12.7**	746,220	746,220	0.00	14.3
12-6-3-L	805,930	810,740	0.48	*	809,840	809,840	0.00	112.6	809,840	809,840	0.00	79.0	809,840	809,840	0.00	**72.5**
10-5-4-L	1,006,635	1,031,100	2.11	*	1,027,592	1,028,320	0.07	*	1,028,194	1,028,320	**0.01**	*	1,027,233	1,028,320	0.11	*
15-10-3-L	1,006,000	1,006,200	0.02	*	1,006,200	1,006,200	0.00	46.4	1,006,200	1,006,200	0.00	27.8	1,006,200	1,006,200	0.00	**25.0**
15-12-3-L	1,001,200	1,002,800	0.16	*	1,002,800	1,002,800	0.00	**1.5**	1,002,800	1,002,800	0.00	**1.5**	1,002,800	1,002,800	0.00	**1.5**
15-10-4-L	1,459,400	1,459,600	0.01	*	1,459,600	1,459,600	0.00	23.8	1,459,600	1,459,600	0.00	40.2	1,459,600	1,459,600	0.00	**13.5**
20-10-3-L	1,341,640	-	0.23	*	1,344,446	1,344,800	0.03	*	1,344,450	1,344,800	0.03	*	1,344,467	1,344,800	0.02	*
20-12-3-L	1,331,640	1,343,000	0.36	*	1,336,400	1,336,400	0.00	**2.1**	1,336,400	1,336,400	0.00	2.2	1,336,400	1,336,400	0.00	2.2
4-3-3-T	318,440	318,440	0.00	0.3	318,440	318,440	0.00	0.7	318,440	318,440	0.00	**0.2**	318,440	318,440	0.00	**0.2**
5-3-3-T	405,240	405,240	0.00	**0.6**	405,240	405,240	0.00	1.5	405,240	405,240	0.00	1.3	405,240	405,240	0.00	1.1
6-3-3-T	510,920	510,920	0.00	4.0	510,920	510,920	0.00	1.9	510,920	510,920	0.00	0.9	510,920	510,920	0.00	**0.8**
6-3-4-T	993,460	993,460	0.00	3.5	993,460	993,460	0.00	**1.2**	993,460	993,460	0.00	1.3	993,460	993,460	0.00	**1.2**
10-4-4-T	1,574,771	1,676,990	5.17	*	1,660,640	1,660,640	0.00	101.3	1,660,640	1,660,640	0.00	63.9	1,660,640	1,660,640	0.00	**61.0**
10-4-3-T	973,445	1,023,890	4.77	*	1,022,200	1,022,200	0.00	12.3	1,022,200	1,022,200	0.00	**6.6**	1,022,200	1,022,200	0.00	7.9
4-4-4-T	442,600	442,600	0.00	0.9	442,600	442,600	0.00	1.1	442,600	442,600	0.00	0.7	442,600	442,600	0.00	**0.6**
5-4-4-T	576,010	576,010	0.00	**4.1**	576,010	576,010	0.00	10.3	576,010	576,010	0.00	6.0	576,010	576,010	0.00	6.6
6-4-4-T	653,560	653,560	0.00	11.5	653,560	653,560	0.00	23.5	653,560	653,560	0.00	**8.8**	653,560	653,560	0.00	10.4
12-5-3-T	811,240	835,740	2.31	*	830,440	830,440	0.00	128.0	830,440	830,440	0.00	**68.3**	830,440	830,440	0.00	96.1
12-6-3-T	805,180	823,240	1.67	*	818,840	818,840	0.00	173.6	818,840	818,840	0.00	163.7	818,840	818,840	0.00	**158.6**
10-5-4-T	1,117,723	1,147,530	2.31	*	1,144,160	1,144,160	0.00	141.4	1,144,160	1,144,160	0.00	**68.6**	1,144,160	1,144,160	0.00	72.6
15-10-4-T	1,575,640	1,605,460	1.34	*	1,597,100	1,597,100	0.00	**9.3**	1,597,100	1,597,100	0.00	11.5	1,597,100	1,597,100	0.00	13.7
20-10-3-T	1,551,597	-	4.78	*	1,629,000	1,629,500	**0.03**	*	1,629,000	1,629,500	0.03	*	1,629,000	1,629,500	0.03	*
20-12-3-T	1,541,949	1,628,900	4.02	*	1,606,500	1,606,500	0.00	46.3	1,606,500	1,606,500	0.00	45.4	1,606,500	1,606,500	0.00	**42.5**
Average			1.113				0.0079				0.0060				0.0081	
Optimal solutions			15/34				30/34				30/34				30/34	

Table 2 Computational results on instances from Venturini et al. (2017) with a total time limit of 3 hours and 18 minutes. The MIP formulation is compared to the variants of the presented branch-and-cut-and-price method. "*" means the time limit has been reached. The best running time is highlighted in bold for instances solved to optimality and the best optimality gap for the rest of instances.

Instance	MIP formulation				Branch & Price				Branch & Cut (root node) & Price				Branch & Cut & Price			
N-B-P-TW	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)
4-3-3-L	296,600	296,600	0.00	**0.1**	296,600	296,600	0.00	0.5	296,600	296,600	0.00	0.2	296,600	296,600	0.00	0.2
5-3-3-L	394,300	394,300	0.00	**0.4**	394,300	394,300	0.00	3.2	394,300	394,300	0.00	7.2	394,300	394,300	0.00	7.2
6-3-3-L	421,679	421,720	0.01	2.3	421,720	421,720	0.00	0.8	421,720	421,720	0.00	**0.5**	421,720	421,720	0.00	**0.5**
6-3-4-L	647,423	647,480	0.01	89.2	647,480	647,480	0.00	**39.0**	647,480	647,480	0.00	111.7	647,480	647,480	0.00	103.7
10-4-4-L	1,020,581	1,055,800	3.22	*	1,054,500	1,054,500	0.00	5563.8	1,054,500	1,054,500	0.00	6068.2	1,054,300	1,054,500	**0.02**	*
10-4-3-L	694,451	699,000	0.52	*	698,100	698,100	0.00	193.6	698,100	698,100	0.00	**72.7**	698,100	698,100	0.00	115.6
4-4-4-L	405,120	405,120	0.00	**0.3**	405,120	405,120	0.00	0.6	405,120	405,120	0.00	0.6	405,120	405,120	0.00	0.6
5-4-4-L	500,600	500,600	0.00	**0.4**	500,600	500,600	0.00	0.9	500,600	500,600	0.00	0.8	500,600	500,600	0.00	0.8
6-4-4-L	599,980	599,980	0.00	**1.2**	599,980	599,980	0.00	7.6	599,980	599,980	0.00	5.1	599,980	599,980	0.00	5.3
12-5-3-L	813,713	834,740	2.01	*	830,440	830,440	0.00	136.0	830,440	830,440	0.00	**112.1**	830,440	830,440	0.00	120.4
10-6-3-L	680,600	680,600	0.00	219.4	680,600	680,600	0.00	9.7	680,600	680,600	0.00	**5.1**	680,600	680,600	0.00	5.8
11-6-3-L	746,220	746,220	0.00	8705.8	746,220	746,220	0.00	22.5	746,220	746,220	0.00	**12.7**	746,220	746,220	0.00	14.3
12-6-3-L	809,840	809,840	0.00	5032.0	809,840	809,840	0.00	112.6	809,840	809,840	0.00	79.0	809,840	809,840	0.00	**72.5**
10-5-4-L	1,013,114	1,029,300	1.48	*	1,028,320	1,028,320	0.00	589.0	1,028,320	1,028,320	0.00	**366.1**	1,028,320	1,028,320	0.00	1135.1
15-10-3-L	1,006,200	1,006,200	0.00	3259.5	1,006,200	1,006,200	0.00	46.4	1,006,200	1,006,200	0.00	27.8	1,006,200	1,006,200	0.00	**25.0**
15-12-3-L	1,002,240	1,002,800	0.06	*	1,002,800	1,002,800	0.00	**1.5**	1,002,800	1,002,800	0.00	**1.5**	1,002,800	1,002,800	0.00	**1.5**
15-10-4-L	1,459,600	1,459,600	0.00	1703.1	1,459,600	1,459,600	0.00	23.8	1,459,600	1,459,600	0.00	40.2	1,459,600	1,459,600	0.00	**13.5**
20-10-3-L	1,341,640	1,346,000	0.23	*	1,344,520	1,344,800	0.02	*	1,344,525	1,344,800	0.02	*	1,344,600	1,344,800	**0.01**	*
20-12-3-L	1,331,680	1,337,400	0.35	*	1,336,400	1,336,400	0.00	**2.1**	1,336,400	1,336,400	0.00	2.2	1,336,400	1,336,400	0.00	2.2
4-3-3-T	318,440	318,440	0.00	0.3	318,440	318,440	0.00	0.7	318,440	318,440	0.00	**0.2**	318,440	318,440	0.00	**0.2**
5-3-3-T	405,240	405,240	0.00	**0.6**	405,240	405,240	0.00	1.5	405,240	405,240	0.00	1.3	405,240	405,240	0.00	1.1
6-3-3-T	510,920	510,920	0.00	4.0	510,920	510,920	0.00	1.9	510,920	510,920	0.00	0.9	510,920	510,920	0.00	**0.8**
6-3-4-T	993,460	993,460	0.00	3.5	993,460	993,460	0.00	**1.2**	993,460	993,460	0.00	1.3	993,460	993,460	0.00	**1.2**
10-4-4-T	1,660,640	1,660,640	0.00	1660.0	1,660,640	1,660,640	0.00	101.3	1,660,640	1,660,640	0.00	63.9	1,660,640	1,660,640	0.00	**61.0**
10-4-3-T	1,022,200	1,022,200	0.00	562.5	1,022,200	1,022,200	0.00	12.3	1,022,200	1,022,200	0.00	**6.6**	1,022,200	1,022,200	0.00	7.9
4-4-4-T	442,600	442,600	0.00	0.9	442,600	442,600	0.00	1.1	442,600	442,600	0.00	0.7	442,600	442,600	0.00	**0.6**
5-4-4-T	576,010	576,010	0.00	**4.1**	576,010	576,010	0.00	10.3	576,010	576,010	0.00	6.0	576,010	576,010	0.00	6.6
6-4-4-T	653,560	653,560	0.00	11.5	653,560	653,560	0.00	23.5	653,560	653,560	0.00	**8.8**	653,560	653,560	0.00	10.4
12-5-3-T	817,533	830,440	1.55	*	830,440	830,440	0.00	128.0	830,440	830,440	0.00	**68.3**	830,440	830,440	0.00	96.1
12-6-3-T	810,476	821,540	1.02	*	818,840	818,840	0.00	173.6	818,840	818,840	0.00	163.7	818,840	818,840	0.00	**158.6**
10-5-4-T	1,144,160	1,144,160	0.00	3649.8	1,144,160	1,144,160	0.00	141.4	1,144,160	1,144,160	0.00	**68.6**	1,144,160	1,144,160	0.00	72.6
15-10-4-T	1,584,071	1,597,620	0.82	*	1,597,100	1,597,100	0.00	**9.3**	1,597,100	1,597,100	0.00	11.5	1,597,100	1,597,100	0.00	13.7
20-10-3-T	1,552,283	1,634,900	4.74	*	1,629,380	1,629,500	**0.01**	*	1,629,300	1,629,500	0.01	*	1,629,380	1,629,500	**0.01**	*
20-12-3-T	1,545,006	1,609,100	3.83	*	1,606,500	1,606,500	0.00	46.3	1,606,500	1,606,500	0.00	45.4	1,606,500	1,606,500	0.00	**42.5**
Average		0.584				0.0008				0.0010				0.0012		
Optimal solutions		24/34				32/34				32/34				31/34		

Table 3 Computational results on the set of harder instances with a total time limit of 5 minutes and 30 seconds. The MIP formulation is compared to the variants of the presented branch-and-cut-and-price method. "-" means that no integer solution has been found within the time limit. "*" means the time limit has been reached. The best running time is highlighted in bold for instances solved to optimality and the best optimality gap for the rest of instances.

Instance	MIP formulation				Branch & Price				Branch & Cut (root node) & Price				Branch & Cut & Price			
N-B-P-TW	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)
25-12-3-L	1,668,080	-	0.56	*	1,677,200	1,677,400	**0.01**	*	1,677,200	1,677,400	**0.01**	*	1,677,200	1,677,400	**0.01**	*
25-12-3-T	1,923,560	-	6.03	*	2,046,930	2,047,100	0.00	*	2,046,933	2,047,200	**0.00**	*	2,046,933	2,047,200	**0.00**	*
12-5-4-L	1,201,546	1,253,160	3.50	*	1,239,277	1,245,360	0.47	*	1,240,826	1,247,760	0.35	*	1,240,862	1,247,060	**0.35**	*
12-5-4-T	1,324,428	-	5.69	*	1,398,848	1,410,270	0.39	*	1,399,631	1,408,070	**0.33**	*	1,398,580	1,408,220	0.41	*
30-12-3-L	1,997,400	-	0.95	*	2,016,178	2,016,600	**0.02**	*	2,016,178	2,016,600	0.02	*	2,016,178	2,016,600	**0.02**	*
30-12-3-T	2,304,432	-	7.55	*	2,491,406	2,496,900	**0.04**	*	2,491,406	2,495,300	0.04	*	2,491,406	2,495,600	**0.04**	*
20-12-4-L	1,934,640	-	0.47	*	1,943,486	1,943,800	0.02	*	1,943,500	1,943,800	0.02	*	1,943,800	1,943,800	0.00	**243.7**
20-12-4-T	3,050,995	-	2.00	*	3,113,170	3,113,170	0.00	42.5	3,113,170	3,113,170	0.00	15.8	3,113,170	3,113,170	0.00	**14.9**
15-8-4-L	1,471,903	1,508,500	1.68	*	1,495,225	1,497,100	0.12	*	1,496,131	1,497,300	**0.06**	*	1,496,118	1,497,100	0.06	*
15-8-4-T	1,599,496	-	3.41	*	1,654,848	1,656,260	0.07	*	1,655,376	1,656,260	0.04	*	1,655,416	1,656,260	**0.04**	*
25-12-4-L	2,419,800	-	0.85	*	2,439,377	2,440,700	0.05	*	2,439,531	2,440,600	**0.04**	*	2,439,380	2,441,500	0.05	*
25-12-4-T	3,560,100	-	3.97	*	3,706,995	3,707,390	0.01	*	3,707,182	3,707,390	0.01	*	3,707,390	3,707,390	**0.00**	*
30-15-4-L	2,905,000	-	0.47	*	2,918,400	2,918,800	0.01	*	2,918,420	2,918,800	0.01	*	2,918,436	2,918,800	**0.01**	*
30-15-4-T	3,109,816	-	5.14	*	3,274,118	3,278,880	0.13	*	3,274,390	3,283,550	0.12	*	3,274,441	3,280,400	**0.12**	*
40-15-3-L	2,658,400	-	1.15	*	2,689,060	2,689,200	**0.01**	*	2,689,060	2,689,200	**0.01**	*	2,689,060	2,689,200	**0.01**	*
40-15-3-T	3,067,200	-	7.89	*	3,329,567	3,330,900	**0.02**	*	3,329,567	3,330,900	**0.02**	*	3,329,567	3,331,300	**0.02**	*
Average			3.206				0.0849				0.0665				0.0698	
Optimal solutions			0/16				1/16				1/16				3/16	

Table 4 Computational results on the set of harder instances with a total time limit of 3 hours and 18 minutes. The MIP formulation is compared to the variants of the presented branch-and-cut-and-price method. "-" means that no integer solution has been found within the time limit. "*" means the time limit has been reached. The best running time is highlighted in bold for instances solved to optimality and the best optimality gap for the rest of instances.

Instance	MIP formulation				Branch & Price				Branch & Cut (root node) & Price				Branch & Cut & Price			
N-B-P-TW	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)
25-12-3-L	1,668,080	1,679,400	0.56	*	1,677,200	1,677,400	**0.01**	*	1,677,200	1,677,400	**0.01**	*	1,677,200	1,677,400	**0.01**	*
25-12-3-T	1,932,150	-	5.61	*	2,046,930	2,047,100	0.00	*	2,046,933	2,047,100	**0.00**	*	2,046,933	2,047,000	**0.00**	*
12-5-4-L	1,205,096	1,253,160	3.22	*	1,244,427	1,245,660	0.06	*	1,245,110	1,245,160	**0.00**	*	1,244,080	1,245,160	0.09	*
12-5-4-T	1,348,176	1,407,980	4.00	*	1,404,280	1,404,280	0.00	**5126.1**	1,404,280	1,404,280	0.00	9530.6	1,403,243	1,405,760	0.07	*
30-12-3-L	1,998,483	-	0.90	*	2,016,178	2,016,600	**0.02**	*	2,016,178	2,016,600	**0.02**	*	2,016,178	2,016,600	**0.02**	*
30-12-3-T	2,311,705	-	7.25	*	2,491,406	2,492,500	**0.04**	*	2,491,406	2,492,500	**0.04**	*	2,491,406	2,492,500	**0.04**	*
20-12-4-L	1,934,640	-	0.47	*	1,943,800	1,943,800	0.00	2544.5	1,943,800	1,943,800	0.00	1202.7	1,943,800	1,943,800	0.00	**243.7**
20-12-4-T	3,055,040	-	1.87	*	3,113,170	3,113,170	0.00	42.5	3,113,170	3,113,170	0.00	15.8	3,113,170	3,113,170	0.00	**14.9**
15-8-4-L	1,475,646	1,497,000	1.43	*	1,497,000	1,497,000	0.00	2870.3	1,497,000	1,497,000	0.00	**968.3**	1,497,000	1,497,000	0.00	1831.1
15-8-4-T	1,608,977	1,671,220	2.84	*	1,656,040	1,656,040	0.00	**746.7.3**	1,656,040	1,656,040	0.00	1365.a	1,656,040	1,656,040	0.00	1534.0
25-12-4-L	2,420,034	-	0.84	*	2,439,594	2,440,500	0.04	*	2,439,796	2,440,500	0.03	*	2,439,881	2,440,500	**0.03**	*
25-12-4-T	3,572,971	-	3.63	*	3,707,390	3,707,390	0.00	604.3	3,707,390	3,707,390	0.00	501.1	3,707,390	3,707,390	0.00	**440.8**
30-15-4-L	2,905,000	-	0.47	*	2,918,400	2,918,800	0.01	*	2,918,425	2,918,800	0.01	*	2,918,441	2,918,600	**0.01**	*
30-15-4-T	3,123,445	-	4.72	*	3,274,905	3,278,280	0.10	*	3,275,095	3,278,280	0.10	*	3,275,112	3,278,940	**0.10**	*
40-15-3-L	2,658,440	-	1.14	*	2,689,060	2,689,200	**0.01**	*	2,689,060	2,689,200	**0.01**	*	2,689,060	2,689,200	**0.01**	*
40-15-3-T	3,071,346	-	7.77	*	3,329,567	3,330,300	**0.02**	*	3,329,567	3,330,200	**0.02**	*	3,329,567	3,330,100	**0.02**	*
Average			2.919				0.0192				0.0148				0.0243	
Optimal solutions			0/16				6/16				6/16				5/16	

Table 5   Performance of presented methods on a subset of five instances.

Instance	Branch & Price										
N-B-P-TW	**Gap (%)**	**T (s)**	**T RMP (%)**	**T PP (%)**	**T Sep (%)**	**T Branch (%)**	**T GSPP (%)**	**Nodes**	**CG Its**	**Cols**	**Cuts**
12-6-3-T	0.00	174	75.8	19.6	0.0	1.5	0.0	219	3,820	16,928	0
10-4-4-L	0.00	5,564	89.2	10.3	0.0	0.1	0.0	233	14,745	91,489	0
20-10-3-L	0.02	11,031	28.7	28.8	0.0	18.7	0.0	13,907	45,145	112,557	0
15-8-4-L	0.00	2,870	48.3	49.8	0.0	0.6	0.0	327	7,553	48,756	0
40-15-3-T	0.02	11,097	74.3	10.5	0.0	2.6	2.1	899	6,678	99,529	0
Instance	**Branch & Cut (root node) & Price**										
N-B-P-TW	**Gap (%)**	**T (s)**	**T RMP (%)**	**T PP (%)**	**T Sep (%)**	**T Branch (%)**	**T GSPP (%)**	**Nodes**	**CG Its**	**Cols**	**Cuts**
12-6-3-T	0.00	164	88.3	9.1	0.4	0.9	0.0	45	1,356	7,215	567
10-4-4-L	0.00	6,068	95.3	4.4	0.1	0.1	0.0	81	5,667	35,916	382
20-10-3-L	0.02	11,089	28.5	26.8	4.1	17.8	0.0	15,421	49,256	131,616	24
15-8-4-L	0.00	968	69.8	27.8	0.4	1.1	0.0	49	1,417	10,629	861
40-15-3-T	0.02	11,022	68.9	12.2	1.4	3.3	1.3	903	6,390	94,066	20
Instance	**Branch & Cut & Price**										
N-B-P-TW	**Gap (%)**	**T (s)**	**T RMP (%)**	**T PP (%)**	**T Sep (%)**	**T Branch (%)**	**T GSPP (%)**	**Nodes**	**CG Its**	**Cols**	**Cuts**
12-6-3-T	0.00	159	85.2	10.9	1.6	1.0	0.0	43	1,426	7,071	1,382
10-4-4-L	0.02	11,120	93.6	3.3	0.2	0.0	2.7	61	6,755	38,497	3,716
20-10-3-L	0.01	11,062	28.0	28.7	11.6	12.9	0.0	12,681	49,949	118,332	32,198
15-8-4-L	0.00	1,831	81.1	17.5	0.6	0.5	0.0	31	1,660	11,727	2,236
40-15-3-T	0.02	11,180	73.1	9.6	6.5	1.2	3.1	515	5,547	79,677	7,124

The results show a better performance of the solution methods based on the GSPP formulation. All variants of the *branch-and-cut-and-price* method are able to find optimal or near-optimal solutions in less than 6 minutes. Among the new proposed methods, the one where cutting is performed at the root node shows a better performance. All the solution method variants outperform CPLEX in all instances that require more than 4 seconds to solve and show similar running times for the faster ones. The difference in performance is more notable on the set of harder instances where CPLEX is not able to find a feasible integer solution in 11 out of the 16 instances within 3 hours and 18 minutes and the average optimality gap is above 2.9 %. Within 5 minutes and 30 seconds, the proposed new methods not only find feasible solutions to all instances but also achieve an optimality gap of 0.03 %. This gap is further reduced to less than 0.01 % with a time limit of 3 hours and 18 minutes. The good quality of the solutions in such a short computational time is attractive from an operational point of view where suboptimal solutions usually are not a problem and possible disruptions require rapid computations for new plans. Regarding the impact of solving the GSPP at the end, it is higher when the time limits are low. The GSPP is only solved if the method still has not proven optimality and, from those cases, it is found that the GSPP improves the upper bound in 84-100% of the cases with the 5 minutes and 30 seconds time limit and in 50-57% of the cases with the 3 hours and 18 minutes, depending on the method. In the vast majority of these cases, an integer solution is not found in the B&B tree.

Table 5 provides a more detailed comparison of the proposed method variants for 5 instances that aim to be representative of the entire set of 50 instances. The first column indicates the instance,

Table 6   Optimality gap to optimal or best known solution at the root node.

Instance	MIP formulation		Without cuts		With sol-based cuts		With all cuts	
N-B-P-TW	Gap (%)	T (s)	Gap (%)	T (s)	Gap (%)	T (s)	Gap (%)	T (s)
12-6-3-T	2.03	0.3	0.36	2	0.22	5	0.22	87
10-4-4-L	4.69	0.1	0.22	15	0.15	102	0.15	1,137
20-10-3-L	0.36	0.2	0.03	2	0.03	2	0.03	143
15-8-4-L	2.11	0.2	0.25	19	0.12	43	0.12	499
40-15-3-T	7.89	1.2	0.02	280	0.02	285	0.02	685
Average 50 instances	**3.97**	**0.3**	**0.23**	**18**	**0.0865**	**25**	**0.0863**	**315**

the second and third column recap the optimality gap and computational time spent given the time-limit of 3 hours and 18 minutes. The fourth to eighth columns indicate the percentage amount of time spent by the algorithm in the RMP, pricing problems (PPs), cut separation process (Sep), branching procedure (Branch), and the final GSPP model respectively. The pricing problems are solved in parallel on the four cores used. It should also be noticed that the branching time not only includes the selection of the branching candidate but also, the child nodes creation, which in the case of our algorithm, requires intensive data structure manipulation. The number of nodes explored in the B&B tree is displayed in the ninth column. The last three columns indicate the number of column generation iterations, generated columns and added cuts respectively.

The RMP takes most of the time for most of the instances, and the cut separation has an insignificant impact except when it is applied in every B&B node. The time spent in branching procedures grows in accordance to both the size of the RMP and B&B tree. The short RMP solving times and large amount of columns generated for instance $20-10-3-L$ suggest that the RMP is easy to solve and the existence of many equivalent or similar solutions. This increases the impact of other internal operations in the algorithm. The number of B&B nodes explored grows inversely to the amount of nodes where cutting is allowed. It can be observed that the full *branch-and-cut-price* performs more column generation iterations than the one with only cutting in the root node but it also requires longer computational times. As it can be observed, the time percentages do not sum exactly to 100%. The remaining time accounts to diverse internal operations in the implementation which are not strictly linked to any of the main parts of the algorithm. This also suggests that there is room for improvement in the implementation of the algorithm.

The effectiveness of the aforementioned cut separation process is displayed in Table 6. The optimality gap of the LP solution at the root node is shown for the subset of five instances studied in detail together with the average across all 50 instances. The second column denotes the LP solution at the root node for the MIP formulation. The third column refers to the presented method without adding any cuts whereas the fourth column considers the proposed cut separation procedure (Algorithm 2) based on solution values (i.e., *sol-based*). This procedure only checks a subset of the valid inequalities which we believe that contains most, if not all, of the violated

Table 7   Solving time comparison between the network flow problem and the branch-and-price method.

Instance	Graph size		Network flow problem	Branch & Price
N-B-P-TW	Nodes	Arcs	T(s)	T(s)
4-3-3-L	1,621	44,989	5.4	0.5
5-3-3-L	2,711	1,669,872	383	3.2
6-3-3-L	2,711	2,353,886	517	0.8
6-3-4-L	5,414	8,659,488	2,504	39.0

Table 8   Average performance of different branching strategies across all 50 instances using the

branch-and-cut-and-price method with only cutting allowed in the root node.

Branching strategy	Best first on single node	Strong branching on berthing time	Best first on berthing position	Best first on berthing time
Average gap (%)	0.058	0.011	0.009	**0.005**
Optimal solutions	16/50	37/50	**39/50**	38/50

ones. In any case, we can find all violated inequalities by simple enumeration. This case, where all violated valid inequalities (i.e., *all cuts*) are added, has also been tested and the results are shown in the last column. The improvement in the lower bound is significant for the proposed methods where the cut separation is able to further improve it achieving an average gap of 0.09 %. Adding all possible cuts only leads to an average improvement of 0.0002 % in the bound. However, the algorithm requires 12 times more computational time to solve the root node. It is therefore decided to discard this variant of the separation procedure given the slow performance and the insignificant gain.

As mentioned in section 4.1, when having a pure shortest path as a pricing problem, solving the LP relaxation of the network flow problem gives the same bound as column generation on the GSPP but the network flow problem is expected to require more time and memory resources on instances with dense graphs. In order to verify that, the network flow problem has been solved for the first four instances which are considered among the *easiest* ones from the entire set. The solving times of the network flow problem and the *branch-and-price* method are compared in Table 7. The complexity of the graph is shown by the high solving times for the network flow problem where the proposed model is between 10 and more than 500 times faster. The rest of instances have not been further analyzed as most of them were reaching the memory limit. The number of nodes and arcs for all the instances are documented in Table 13 in Appendix C.

Apart from the presented methods, slight variations have been tested which helped to select the best algorithm procedure. For instance, we have tried to generate all columns *a priori* without success. The complexity of the problem and the exponentially large numbers of columns make it intractable. Regarding branching procedures, an alternative method of exploring the B&B tree known as *strong branching* has been tested. This strategy requires to select a number of candidates (between 5 and 10 in our case) and compute, or at least estimate, the lower bounds at the child

nodes. For each candidate, a weighted sum of the child bounds is computed and the candidate with the best weighted sum is selected. In this case, a weight of 0.75 is set for the child with the lowest bound and a weight of 0.25 for the other child. This method has proven to create better branches and, for example, is able to find an optimal solution to instance *20-10-3-T* in less than 20 minutes. Nevertheless, the overall worse optimality gap and additional time consumed exploring more nodes has lead us to discard it. A different branching strategy has been tested where the branching is done on berthing positions instead of on berthing times at a port. This strategy is able to solve all the instances from Venturini et al. (2017) to optimality and one more instance in overall than the presented method. However, the overall worse optimality gap indicates a poorer performance on the set of harder instances (see Table 14 in Appendix C). In addition, a trivial branching on a single node has also been tested to compare the effectiveness of the proposed branching strategy. The solution values of the columns are added on the graph nodes of the respective paths, computing in this way the "usage" of each graph node. Then, the graph node whose value is closer to 0.5 (i.e., *most fractional*) is the one selected to branch on. A summary of the performance of these alternative branching strategies is displayed in Table 8 and the results for all instances can be found in Table 14 in Appendix C.

6.2. Cooperative game theory results

The two methods for allocating the costs have been tested in the same set of instances. Three

Table 9   Carrier ship share and priority for the instances.

Carrier	A	B	C
% of ships	50	25	25
Priority	1	2	3

carriers A, B and C have been defined for all instances each of them with an assigned priority and a number of ships (see Table 9). This priority is often given in accordance to the handling volume (Imai, Nishimura, and Papadimitriou 2003). For instance, carrier A can be seen as a large carrier and often this translates in more power of decision and a higher priority at the port. The

Table 10   Terminal denomination.

Terminal	D	E	F	G
Visit position	1	2	3	4

terminal operator at each visited port is also a player in the game. In this study, all ships visit the ports in the same order as shown in Figure 10, but the game can equally be applied to instances with different visit orders. Thus, depending on the instance's number of ports, the game is formed

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

39

by either 6 or 7 players. The number of possible coalitions is given by $2^{|\mathcal{P}|}$ where $|\mathcal{P}|$ denotes the number of players, and in this case, corresponds to 64 or 128 coalitions respectively.

The overall cooperative game is based on what we denote as the *standalone solution*. This solution reflects the scenario where a single carrier negotiates with a single terminal at a time in order to decide the schedule for the carriers' vessels. We apply a greedy heuristic to compute this solution where we optimize and fix the schedule of a carrier's ships one port at a time. The sequence of carriers and ports used by the heuristic is given by the carrier's priority at port (see Table 9) and the position of the port visited (see Table 10). For example, assume ship 1 is carrier A's only ship and visits first port D and then port E where it has the highest priority. We then optimize the schedule of ship 1 for port D, fix the decisions and then optimize the schedule of ship 1 for port E. Once the port visits of a carrier's ships are scheduled, the ships of the next carrier with the highest priority are scheduled within the remaining available berthing positions and time windows.

This sequential planning approach resembles the actual procedure in some ports when the carriers book the port calls and they are assigned based on different priority schemes. Due to the heuristic nature of the process, some of the carriers may not find a feasible schedule. In order to avoid this and still ensure a fair comparison, the operational time windows of all berthing positions have been increased by 20% in the tests performed in this section.

The different coalitions $\mathcal{S} \subseteq \mathcal{P}$ can be classified into three groups, depending on the type of players forming it:

• Coalitions formed by carriers only. For the problem at hand this form of collaboration does not provide any planning advantage as the carriers require collaboration with the terminal operators to improve their planning. Therefore, the solution of this type of coalition corresponds to the standalone solution.

• Coalitions formed by terminal operators only. For the problem at hand this form of collaboration does not provide any planning advantage as the terminal operators require collaboration with the carriers to improve their planning. Therefore, the solution of this type of coalition corresponds to the standalone solution.

• Coalitions formed by both carriers and terminal operators. This type of coalition is the basis for the MPBAP. In order to compute a solution to a given coalition, we assume that the planning of all players that are not part of the coalition are kept fixed as in the standalone solution. Then, the MPBAP is solved for the coalition given the available berthing space and time at the terminals. Note that when more players are part of the coalition, fewer port calls of the standalone solution need to be fixed. In addition, it can be noticed that optimal solutions to coalitions formed by a single carrier and a single operator are equivalent to the standalone solution. We believe this minimal collaboration resembles the real-life port call booking process for carriers.

Table 11   Comparison of the two cost allocation methods across instances with three ports.

	Player	Cost	Shapley value			Equal profit method (EPM)		
	$\mathcal{S}$	$\vartheta(\mathcal{S})$	f_i	Relative savings (%)	% of total costs	f_i	Relative savings (%)	% of total costs
Carrier	**A**	485,261	478,464	1.5	39.1	471,601	3.3	38.4
	B	247,053	242,800	2.2	20.0	240,685	3.1	19.8
	C	251,485	246,914	2.4	20.9	245,092	3.1	20.7
Terminal	**D**	88,635	79,042	10.7	7.0	86,058	3.4	7.5
	E	75,221	70,789	6.9	6.4	73,059	3.3	6.7
	F	75,296	71,546	5.8	6.6	73,060	3.2	6.8
	Grand coalition	1,189,555						

Table 12   Comparison of the two cost allocation methods across instances with four ports.

	Player	Cost	Shapley value			Equal profit method (EPM)		
	$\mathcal{S}$	$\vartheta(\mathcal{S})$	f_i	Relative savings (%)	% of total costs	f_i	Relative savings (%)	% of total costs
Carrier	**A**	352,695	347,232	1.7	37.7	339,450	4.0	36.8
	B	187,320	181,789	3.1	20.5	179,622	4.3	20.2
	C	192,634	187,148	3.1	20.9	184,727	4.4	20.6
Terminal	**D**	46,264	40,908	11.7	4.6	44,464	4.0	5.0
	E	44,079	38,207	14.1	4.1	42,215	4.4	4.6
	F	66,496	60,718	8.5	6.3	63,629	3.9	6.6
	G	60,346	55,765	7.6	5.9	57,659	3.9	6.1
	Grand coalition	911,766						

The premise of fixing the port calls of non-collaborators, ensures that, in the worst case, the standalone solution is feasible for any coalition $\mathcal{S} \subseteq \mathcal{P}$. As indicated in Section 3.2, carriers and terminal operators have different operational costs. This is also reflected in the characteristic function, where the costs of each player are measured differently. On one side, the carrier's cost comprise the fuel consumption costs, waiting time costs and half of the delay costs. On the other side, the terminal operator's cost comprise the handling costs and the remaining half of the delay costs. The process of quantifying the delay costs in this type of problems is complex and it has been decided to equally split the delay costs between carriers and terminal operators.

As shown in Section 6.1, the grand coalition scenario for some instances are not solved to optimality, but the proposed methods are able to find solutions within a very small gap in less than 6 minutes (see Tables 1 and 3). It is assumed, that all the subcoalition scenarios are at most, as hard to solve as the grand coalition one and, therefore, a time limit of 5 minutes, and an additional 30 seconds to solve the GSPP, is set to solve each coalition scenario of the game.

All instances have a non-empty core, meaning that both efficient and stable solutions can be found in all scenarios. Tables 11 and 12 show the average cost allocations to each of the carriers and terminal operators across instances with 3 and 4 ports respectively (24 out of the 50 instances

have 4 ports). For each allocation method we display three columns, (i) the first column indicates the cost allocation to the carrier when being part of the grand coalition, (ii) the second column computes the percentual savings compared with the player's standalone cost and (iii) the third column shows the percentage of the overall costs allocated to each player. Both allocation methods show that significant savings can be achieved by all of the players involved. In fact, player A, which in theory may be the least interested in engaging in such grand coalitions due to its high priority at all ports, achieves significant savings. The same applies to terminal D, which is the first one visited by the ships, and can benefit substantially by the overall better planning of the rest of terminals. The differences in the allocation strategy used by the two methods are noticeable. The EPM tends to equalize the relative savings of all players whereas the Shapley value is prone to balance the absolute savings.

As mentioned in Section 1, we conceive that the solution to the MPBAP and the cost allocation methods could be provided as a service by third party software companies. To establish the side payments in practice, players would need to commit to the service for a pre-established period and agree with the potential savings estimated by the third party. This is required in order to define the number of participants in the collaboration. Moreover, to measure the savings of the MPBAP solution, we need an estimate of the standalone costs of each player in a non-collaborative scenario. The third party could estimate this cost for each player using the current planning software and we assume that the player would agree to that estimate. Once the agreement is in place, the third party could be used as a proxy for the side payments, which could be performed on a regular basis. Based on the actual costs, each player would need to make or receive a payment to align with the projected savings that the player has agreed to.

Similar collaboration is already taking place in real-life in tramp shipping where it is common that a number of ship owners place their ships in a shipping pool under the control of a pool administrator (the analog to the third party coordinator) that takes over most of the business decisions regarding the ships and is responsible for distributing earnings to ship owners (Packard 1995, Wen et al. 2019).

7. Conclusions and future work

A novel solution method based on a GSPP formulation has been presented for the MPBAP. The method exploits a graph formulation for defining the berthing plan of a ship along its route. This, combined together with delayed column generation, additional valid inequalities and symmetry breaking constraints results in an efficient algorithm able to find optimal or near-optimal solutions to wide range of instances outperforming the capacity of commercial solvers.

In addition, the graph formulation adds flexibility as many additional constraints can be easily integrated with simple alterations in the graph. For instance, a finer discretization of the berthing

positions would allow to approximate the continuous version of the MPBAP better. Considering a continuous berth is a more realistic approach and allows to increase the usage of the quay. Transhipments are also an important aspect of the operations at port and the fulfillment of them are crucial in some cases (e.g., when transporting perishable food). Our model could eventually account for that by limiting the time window of the ships involved in the transhipment and penalizing late arrivals of the incoming ship or too early departures of the outgoing ship. Nevertheless, this could be better modelled if the relative arrival and departure times are considered. That case is harder to incorporate in the presented model and it would require additional constraints for each transhipment in the RMP. The transit times between ports could be further improved by considering the time needed to enter and leave the port which is usually performed at a slower speed (Reinhardt et al. 2016).

The instances solved reflect the size of real-life scenarios to a large extent. However, some of the instance parameters could be further improved. This comprises improving the size of the vessel time windows, having different routes for the ships, different amount of berthing positions per port and different ship types.

Alternative branching methods have also shown great potential, especially branching on berthing positions as opposed to branching on berthing times. A natural next step would be to explore a branching method that combines both of them. For instance, one could test both methods simultaneously when branching and select the one with better bounds at the children nodes.

A natural extension of the problem could be to integrate the berth allocation with the quay crane assignment problem (QAP). Studies such as Iris et al. (2015) and Iris, Pacino, and Røpke (2017) have already shown the effectiveness of heuristic and exact methods based on a GSPP formulation for the integrated problem in one terminal.

Last but not least, the benefits for both ship carriers and terminal operators are verified defining a cooperative game and using cost allocation methods to distribute the costs of such collaboration fairly. The results of the cooperative game strengthen the viability of such a decision tool and can encourage carriers and port operators to participate in collaborative schemes.

Acknowledgments

Appendix A: Reduced cost computation including valid inequalities (33)

We denote $\beta_{t_1,t_2}^{n,p,b}$ to the dual variable of constraint (33) for ship $n \in N$, port $p \in P$, berth $b \in B_p$ and times $t_1, t_2 \in [s^{p,b}; e^{p,b}], t_1 < t_2$ and let $\bar{\beta}_{t_1,t_2}^{n,p,b}$ be its value for the RMP solution. Let $w(p,b,t) \in O$ be the graph node related to berthing at port $p \in P$ at position $b \in B_p$ at time $t \in [s^{p,b}; e^{p,b}]$ and let $\Phi = \{(n,p,b,t_1,t_2)\}$ be the set of constraints (33) added to the RMP denoted by (n,p,b,t_1,t_2) elements where $n \in N$, $p \in P$, $b \in B_p$ and $t_1, t_2 \in [s^{p,b}; e^{p,b}], t_1 < t_2$. Additionally, let $\Phi(k,p,b,t) \subseteq \Phi$ be the set of valid inequalities that include arcs from the graph node $w(p,b,t)$ for a given ship k. By definition the range of nodes for each ship within a valid inequality differ if $k = n$ or $k \neq n$. Therefore we denote $\Phi_{k=n}(k,p,b,t), \Phi_{k \neq n}(k,p,b,t) \subseteq \Phi(k,p,b,t)$ to the subset of cuts (n,p,b,t_1,t_2) where $k = n$ and $k \neq n$ respectively, that together form the entire set $\Phi(k,p,b,t) = \Phi_{k=n}(k,p,b,t) \cup \Phi_{k \neq n}(k,p,b,t)$ and are defined mathematically as follows:

$$\Phi_{k=n}(k,p,b,t) = \left\{ (n,p,b,t_1,t_2) | k = n, w(p,b,t) \in \bigcup_{t \in [t_1;t_2]} C(k,p,b,t) \right\}$$

$$\Phi_{k \neq n}(k,p,b,t) = \left\{ (n,p,b,t_1,t_2) | k \neq n, w(p,b,t) \in C(k,p,b,t_1) \cap C(k,p,b,t_2) \right\}$$

The computation of the reduced cost $\hat{c}_j$ for column j of ship $k \in N$ is updated as follows:

$$\hat{c}_j = c_j - \left(\sum_{(p,b,t) \in \Lambda_j} \left(\sum_{t' \in [t;t+h_k^{p,b})} \bar{\mu}_{p,b,t'} \right) - \left(\sum_{(n,p,b,t_1,t_2) \in \Phi(k,p,b,t)} \bar{\beta}_{t_1,t_2}^{n,p,b} \right) \right) - \bar{\alpha}_k$$

Appendix B: Adaption of the proposed valid inequality considering berth types

PROPOSITION 2. *Given two time instants $t_1, t_2 \in [s^{p,k}, e^{p,k})$ where $t_1 < t_2$ and a port $p \in P$, berth type $k \in K_p$ and ship $n \in N$, the following is a valid inequality:*

$$\sum_{u \in \bigcup_{t \in [t_1;t_2]} C(n,p,k,t)} \sum_{w \in \delta_n^+(u)} x_{u,w}^n + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,k,t_1) \cap C(m,p,k,t_2)} \sum_{w \in \delta_m^+(u)} x_{u,w}^m \leqslant \beta^k$$

Proof. Constraint (34) has been adapted from constraint (30) which is a direct translation from constraint (26) from the network-flow formulation. Therefore, constraint (34) can be formulated as a network-flow problem constraint as follows:

$$\sum_{m \in N} \sum_{i \in C(m,p,k,t)} \sum_{j \in \delta_m^+(i)} x_{i,j}^m \leq \beta^k \quad \forall p \in P, k \in K_p, t \in [s^{p,k}; e^{p,k})$$

where the $C(m,p,k,t)$ defines the set of nodes for ship $m \in N$ that are in conflict with time $t \in [s^{p,k}; e^{p,k})$ in berth type $k \in K_p$ of port $p \in P$. Based on this definition, the intersection set $C(m,p,k,t_1) \cap C(m,p,k,t_2)$ directly defines the set of nodes for ship m that are in conflict with both time instants t_1 and t_2. Constraint (34) indicates that at most β^k (i.e., number of berths of type $k \in B_k$) arcs can be chosen out of the nodes from the sets $C(m,p,k,t)$ of all ships $m \in N$ and, therefore, the same applies to the intersection sets $C(m,p,k,t_1) \cap C(m,p,k,t_2)$. Based on the premise that each ship can only berth in one position, we can relax the requirement of being in conflict with both t_1 and t_2 for a single ship n and only require it to be in conflict with t_1 or t_2. In practice, this means, on one hand, that if ship n berths at a period covering t_1 or t_2, then, at most $\beta^k - 1$ ships $m \in N \setminus \{n\}$ can berth in a period covering both t_1 and t_2. On the other hand, if β^k ships $m \in N \setminus \{n\}$ are berthing at times whose periods cover t_1 and t_2, then ship n is not able to berth at a period covering

t_1 or t_2. The relaxed node interval for ship n can be defined as the union of $C(n,p,k,t_1) \cup C(n,p,k,t_2)$. Considering these node sets, we can define the following valid inequality:

$$\sum_{u \in C(n,p,k,t_1) \cup C(n,p,k,t_2)} \sum_{w \in \delta_n^+(u)} x_{u,w}^n + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,k,t_1) \cap C(m,p,k,t_2)} \sum_{w \in \delta_m^+(u)} x_{u,w}^m \leqslant \beta^k$$

$$\forall p \in P, k \in K_p, n \in N, t_1, t_2 \in [s^{p,k}, e^{p,k}), t_1 < t_2$$

Based on the assumption that a berthing period cannot be discontinued, the intersection set $C(m,p,k,t_1) \cap C(m,p,k,t_2)$ for any ship is not only in conflict with times t_1 and t_2 but with all the time instants in the period $[t_1; t_2]$. Therefore the interval for ship n can be expanded to the union of $C(n,p,k,t)$ sets for all time instants $t \in [t_1; t_2]$ and the resulting valid inequality can be formulated as follows:

$$\sum_{u \in \bigcup_{t \in [t_1;t_2]} C(n,p,k,t)} \sum_{w \in \delta_n^+(u)} x_{u,w}^n + \sum_{m \in N \setminus \{n\}} \sum_{u \in C(m,p,k,t_1) \cap C(m,p,k,t_2)} \sum_{w \in \delta_m^+(u)} x_{u,w}^m \leqslant \beta^k$$

$$\forall p \in P, k \in K_p, n \in N, t_1, t_2 \in [s^{p,k}, e^{p,k}), t_1 < t_2$$

□

Appendix C: Additional computational results

References

APM Terminals, 2021 *APM Terminals.* https://www.apmterminals.com/, accessed: 2021-09-15.

Bektaş T, Ehmke JF, Psaraftis HN, Puchinger J, 2019 *The role of operational research in green freight transportation. European Journal of Operational Research* 274(3):807–823, URL http://dx.doi.org/10.1016/j.ejor.2018.06.001.

Bezanson J, Edelman A, Karpinski S, Shah VB, 2017 *Julia: A fresh approach to numerical computing. Siam Review* 59(1):65–98, URL http://dx.doi.org/10.1137/141000671.

Bierwirth C, Meisel F, 2015 *A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research* 244(3):12689, 675–689, URL http://dx.doi.org/10.1016/j.ejor.2014.12.030.

Brouer BD, Pisinger D, Spoorendonk S, 2011 *Liner shipping cargo allocation with repositioning of empty containers. INFOR Journal* 49(2):109–124, URL http://dx.doi.org/10.3138/infor.49.2.109.

Buhrkal KF, Zuglian S, Røpke S, Larsen J, Lusby RM, 2011 *Models for the discrete berth allocation problem: A computational comparison. Transportation Research. Part E: Logistics and Transportation Review* 47(4):461–473, URL http://dx.doi.org/10.1016/j.tre.2010.11.016.

Carlo HJ, Vis IF, Roodbergen KJ, 2014 *Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. European Journal of Operational Research* 236(1):1–13, URL http://dx.doi.org/10.1016/j.ejor.2013.11.023.

Chang YT, Tongzon J, Luo M, Lee PTW, 2012 *Estimation of optimal handling capacity of a container port: An economic approach. Transport Reviews* 32(2):241–258, URL http://dx.doi.org/10.1080/01441647.2011.644346.

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

45

Table 13 **Number of nodes and arcs in graph** G **for each of the instances. An horizontal line is used to indicate the separation between the set of benchmark instances by Venturini et al. (2017) and the newly generated set of harder instances.**

Instance	Graph size		Instance	Graph size	
N-B-P-TW	Nodes	Arcs	N-B-P-TW	Nodes	Arcs
4-3-3-L	1,621	44,989	4-4-4-T	3,948	1,239,853
5-3-3-L	2,711	1,669,872	5-4-4-T	3,948	1,498,010
6-3-3-L	2,711	2,353,886	6-4-4-T	3,948	1,318,353
6-3-4-L	5,414	8,659,488	12-5-3-T	3,217	2,572,191
10-4-4-L	7,218	25,069,050	12-6-3-T	3,860	3,586,436
10-4-3-L	3,614	6,583,065	10-5-4-T	5,722	8,069,850
4-4-4-L	6,418	5,631,290	15-10-4-T	8,870	25,003,591
5-4-4-L	6,418	8,436,182	20-10-3-T	5,120	11,456,682
6-4-4-L	6,418	10,062,500	20-12-3-T	6,826	21,299,213
12-5-3-L	4,517	12,072,289	25-12-3-L	7,226	65,709,632
10-6-3-L	5,420	15,698,842	25-12-3-T	6,826	26,586,128
11-6-3-L	5,420	15,640,325	12-5-4-L	9,022	45,812,954
12-6-3-L	5,420	17,117,528	12-5-4-T	5,722	9,550,769
10-5-4-L	9,022	38,507,345	30-12-3-L	7,226	78,917,800
15-10-3-L	5,420	21,596,125	30-12-3-T	6,826	31,947,951
15-12-3-L	7,226	37,072,022	20-12-4-L	15,638	171,445,244
15-10-4-L	12,430	76,196,427	20-12-4-T	11,158	56,721,679
20-10-3-L	5,420	28,790,640	15-8-4-L	14,434	144,240,102
20-12-3-L	7,226	52,608,061	15-8-4-T	9,154	29,624,264
4-3-3-T	1,621	229,372	25-12-4-L	15,638	214,550,093
5-3-3-T	2,711	2,696,410	25-12-4-T	11,158	70,841,345
6-3-3-T	2,711	2,538,662	30-15-4-L	16,541	300,541,288
6-3-4-T	3,464	2,307,063	30-15-4-T	11,801	98,992,726
10-4-4-T	4,618	6,633,974	40-15-3-L	8,129	133,728,684
10-4-3-T	2,614	1,825,953	40-15-3-T	7,679	54,138,993

Cheong CY, Tan KC, Liu DK, Lin CJ, 2010 *Multi-objective and prioritized berth allocation in containner ports. Annals of Operations Research* 180(1):63–103, URL http://dx.doi.org/10.1007/s10479-008-0493-0.

Cordeau JF, Laporte G, Legato P, Moccia L, 2005 *Models and tabu search heuristics for the berth-allocation problem. Transportation Science* 39(4):526–538, URL http://dx.doi.org/10.1287/trsc.1050.0120.

Cormen T, Leiserson C, Rivest R, 1996 *Introduction to algorithms* (MIT Press,), ISBN 0070131430.

Corry P, Bierwirth C, 2019 *The berth allocation problem with channel restrictions. Transportation Science* 53(3):708–727, URL http://dx.doi.org/10.1287/trsc.2018.0865.

Dantzig G, Wolfe P, 1960 *Decomposition principle for linear-programs. Operations Research* 8(1):101–111, URL http://dx.doi.org/10.1287/opre.8.1.101.

Du Y, Chen Q, Lam JSL, Xu Y, Cao JX, 2015 *Modeling the impacts of tides and the virtual arrival policy in berth allocation. Transportation Science* 49(4):939–956, URL http://dx.doi.org/10.1287/trsc.2014.0568.

Table 14   **Results of solution methods with alternative branching strategies. The underlying algorithm is a branch-and-cut-and-price where cutting is only allowed at the root node. "*" means the time limit of 3 hours and 18 minutes has been reached.**

Instance	Best first on single graph node				Strong branching on berthing time				Best first on berthing position			
N-B-P-TW	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)	LB	Z	Gap (%)	T (s)
4-3-3-L	296,600	296,600	0.00	0.2	296,600	296,600	0.00	0.2	296,600	296,600	0.00	0.2
5-3-3-L	394,300	394,300	0.00	8.9	394,300	394,300	0.00	9.2	394,300	394,300	0.00	8.1
6-3-3-L	421,720	421,720	0.00	0.5	421,720	421,720	0.00	0.6	421,720	421,720	0.00	0.5
6-3-4-L	647,149	647,480	0.05	*	647,480	647,480	0.00	1,024.4	647,480	647,480	0.00	93.7
10-4-4-L	1,053,050	1,054,500	0.14	*	1,054,250	1,054,500	0.02	*	1,054,500	1,054,500	0.00	9,628.2
10-4-3-L	697,057	698,100	0.15	*	698,100	698,100	0.00	443.5	698,100	698,100	0.00	173.0
4-4-4-L	405,120	405,120	0.00	0.9	405,120	405,120	0.00	0.7	405,120	405,120	0.00	0.6
5-4-4-L	500,600	500,600	0.00	1.2	500,600	500,600	0.00	0.9	500,600	500,600	0.00	0.9
6-4-4-L	599,780	599,980	0.03	*	599,980	599,980	0.00	26.9	599,980	599,980	0.00	7.4
12-5-3-L	829,897	830,440	0.07	*	830,440	830,440	0.00	646.1	830,440	830,440	0.00	143.3
10-6-3-L	680,550	680,600	0.01	*	680,600	680,600	0.00	34.4	680,600	680,600	0.00	15.1
11-6-3-L	745,960	746,220	0.03	*	746,220	746,220	0.00	52.4	746,220	746,220	0.00	16.9
12-6-3-L	809,093	810,040	0.09	*	809,840	809,840	0.00	301.7	809,840	809,840	0.00	54.3
10-5-4-L	1,026,521	1,028,320	0.17	*	1,028,320	1,028,320	0.00	2,472.5	1,028,320	1,028,320	0.00	5,991.3
15-10-3-L	1,006,000	1,006,200	0.02	*	1,006,200	1,006,200	0.00	59.8	1,006,200	1,006,200	0.00	4.8
15-12-3-L	1,002,800	1,002,800	0.00	2.0	1,002,800	1,002,800	0.00	1.8	1,002,800	1,002,800	0.00	2.3
15-10-4-L	1,459,600	1,459,600	0.00	339.9	1,459,600	1,459,600	0.00	62.6	1,459,600	1,459,600	0.00	14.3
20-10-3-L	1,344,437	1,344,800	0.03	*	1,344,583	1,344,800	0.02	*	1,344,800	1,344,800	0.00	163.3
20-12-3-L	1,336,400	1,336,400	0.00	3.4	1,336,400	1,336,400	0.00	2.9	1,336,400	1,336,400	0.00	2.2
4-3-3-T	318,440	318,440	0.00	0.5	318,440	318,440	0.00	0.3	318,440	318,440	0.00	0.3
5-3-3-T	405,240	405,240	0.00	2.5	405,240	405,240	0.00	3.6	405,240	405,240	0.00	7.4
6-3-3-T	510,920	510,920	0.00	1.5	510,920	510,920	0.00	1.1	510,920	510,920	0.00	0.8
6-3-4-T	993,460	993,460	0.00	2.5	993,460	993,460	0.00	2.5	993,460	993,460	0.00	1.2
10-4-4-T	1,660,640	1,660,640	0.00	1,193.6	1,660,640	1,660,640	0.00	329.9	1,660,640	1,660,640	0.00	132.9
10-4-3-T	1,022,200	1,022,200	0.00	49.1	1,022,200	1,022,200	0.00	56.8	1,022,200	1,022,200	0.00	9.1
4-4-4-T	442,600	442,600	0.00	1.3	442,600	442,600	0.00	1.0	442,600	442,600	0.00	0.8
5-4-4-T	575,350	576,010	0.11	*	576,010	576,010	0.00	26.7	576,010	576,010	0.00	9.6
6-4-4-T	651,480	654,040	0.32	*	653,560	653,560	0.00	39.0	653,560	653,560	0.00	10.6
12-5-3-T	830,067	830,440	0.04	*	830,440	830,440	0.00	355.7	830,440	830,440	0.00	100.0
12-6-3-T	817,602	819,040	0.15	*	818,840	818,840	0.00	369.4	818,840	818,840	0.00	141.1
10-5-4-T	1,143,431	1,144,160	0.06	*	1,144,160	1,144,160	0.00	218.5	1,144,160	1,144,160	0.00	53.0
15-10-4-T	1,596,310	1,597,100	0.05	*	1,597,100	1,597,100	0.00	66.0	1,597,100	1,597,100	0.00	21.1
20-10-3-T	1,629,000	1,629,500	0.03	*	1,629,500	1,629,500	0.00	1,138.7	1,629,500	1,629,500	0.00	408.7
20-12-3-T	1,606,500	1,606,500	0.00	50.5	1,606,500	1,606,500	0.00	80.1	1,606,500	1,606,500	0.00	38.3
Average			**0.0461**				**0.0012**				**0.0000**	
25-12-3-L	1,677,200	1,677,400	0.01	*	1,677,234	1,677,400	0.01	*	1,677,233	1,677,400	0.01	*
25-12-3-T	2,046,933	2,047,000	0.00	*	2,046,933	2,047,000	0.00	*	2,046,933	2,047,000	0.00	*
12-5-4-L	1,240,248	1,245,160	0.39	*	1,243,262	1,245,160	0.15	*	1,243,019	1,245,160	0.17	*
12-5-4-T	1,398,454	1,404,640	0.41	*	1,402,329	1,404,520	0.14	*	1,403,550	1,405,580	0.05	*
30-12-3-L	2,016,178	2,016,600	0.02	*	2,016,178	2,016,600	0.02	*	2,016,245	2,016,600	0.02	*
30-12-3-T	2,491,406	2,492,500	0.04	*	2,491,490	2,492,600	0.04	*	2,491,437	2,492,600	0.04	*
20-12-4-L	1,943,545	1,943,800	0.01	*	1,943,800	1,943,800	0.00	778.0	1,943,800	1,943,800	0.00	635.2
20-12-4-T	3,113,085	3,113,170	0.00	*	3,113,170	3,113,170	0.00	75.1	3,113,170	3,113,170	0.00	40.8
15-8-4-L	1,495,243	1,497,100	0.12	*	1,497,000	1,497,000	0.00	3,228.6	1,497,000	1,497,000	0.00	4,309.9
15-8-4-T	1,655,075	1,656,040	0.06	*	1,656,040	1,656,040	0.00	4,356.3	1,656,040	1,656,040	0.00	792.3
25-12-4-L	2,439,300	2,440,500	0.05	*	2,439,810	2,440,500	0.03	*	2,439,585	2,440,800	0.04	*
25-12-4-T	3,705,997	3,707,790	0.04	*	3,707,390	3,707,390	0.00	578.0	3,707,390	3,707,390	0.00	6,483.4
30-15-4-L	2,918,409	2,918,800	0.01	*	2,918,510	2,918,600	0.00	*	2,918,464	2,918,800	0.00	*
30-15-4-T	3,274,228	3,278,860	0.12	*	3,275,538	3,278,480	0.08	*	3,275,004	3,278,280	0.10	*
40-15-3-L	2,689,060	2,689,200	0.01	*	2,689,060	2,689,200	0.01	*	2,689,060	2,689,200	0.01	*
40-15-3-T	3,329,567	3,330,500	0.02	*	3,329,567	3,330,500	0.02	*	3,329,567	3,330,300	0.02	*
Average			**0.0824**				**0.0314**				**0.0288**	
Optimal solutions			**16/50**				**37/50**				**39/50**	

Du Y, Chen Q, Quan X, Long L, Fung RY, 2011 *Berth allocation considering fuel consumption and vessel emissions. Transportation Research Part E: Logistics and Transportation Review* 47(6):1021–1037, URL http://dx.doi.org/10.1016/j.tre.2011.05.011.

Dulebenets MA, 2018 *A comprehensive multi-objective optimization model for the vessel scheduling problem in liner shipping. International Journal of Production Economics* 196:293–318, URL http://dx.doi.org/10.1016/j.ijpe.2017.10.027.

Dulebenets MA, 2019 *Minimizing the total liner shipping route service costs via application of an efficient collaborative agreement. Ieee Transactions on Intelligent Transportation Systems* 20(1):8315131, URL http://dx.doi.org/10.1109/TITS.2018.2801823.

Dulebenets MA, Golias MM, Mishra S, 2018 *A collaborative agreement for berth allocation under excessive demand. Engineering Applications of Artificial Intelligence* 69:76–92, URL http://dx.doi.org/10.1016/j.engappai.2017.11.009.

Dulebenets MA, Pasha J, Abioye OF, Kavoosi M, 2019 *Vessel scheduling in liner shipping: a critical literature review and future research needs. Flexible Services and Manufacturing Journal* 33(1):43–106, URL http://dx.doi.org/10.1007/s10696-019-09367-2.

Dunning I, Huchette J, Lubin M, 2017 *Jump: A modeling language for mathematical optimization. SIAM Review* 59(2):295–320, URL http://dx.doi.org/10.1137/15M1020575.

Ergun O, Kuyzu G, Savelsbergh M, 2007 *Reducing truckload transportation costs through collaboration. Transportation Science* 41(2):206–221, URL http://dx.doi.org/10.1287/trsc.1060.0169.

Fagerholt K, 2001 *Ship scheduling with soft time windows: An optimisation based approach. European Journal of Operational Research* 131(3):559–571, URL http://dx.doi.org/10.1016/S0377-2217(00)00098-9.

Fagerholt K, Gausel NT, Rakke JG, Psaraftis HN, 2015 *Maritime routing and speed optimization with emission control areas. Transportation Research, Part C: Emerging Technologies* 52:57–73, URL http://dx.doi.org/10.1016/j.trc.2014.12.010.

Fagerholt K, Laporte G, Norstad I, 2010 *Reducing fuel emissions by optimizing speed on shipping routes. Journal of the Operational Research Society* 61(3):523–529, URL http://dx.doi.org/10.1057/jors.2009.77.

Fagerholt K, Psaraftis HN, 2015 *On two speed optimization problems for ships that sail in and out of emission control areas. Transportation Research. Part D: Transport and Environment* 39:56–64, URL http://dx.doi.org/10.1016/j.trd.2015.06.005.

Frisk M, Göthe-Lundgren M, Jörnsten K, Rönnqvist M, 2010 *Cost allocation in collaborative forest transportation. European Journal of Operational Research* 205(2):448–458, URL http://dx.doi.org/10.1016/j.ejor.2010.01.015.

Guan Y, Cheung RK, 2005 *The berth allocation problem: Models and solution methods. Container Terminals and Automated Transport Systems: Logistics Control Issues and Quantitative Decision Support* 141–158, URL http://dx.doi.org/10.1007/3-540-26686-0_6.

Hansen P, Oguz C, 2003 *A note on formulation of the static and dynamic berth allocation problems. Les Cahiers Du Gerad* 30:1–17.

Imai A, Nishimura E, Papadimitriou S, 2003 *Berth allocation with service priority. Transportation Research Part B: Methodological* 37(5):437–457, URL http://dx.doi.org/10.1016/S0191-2615(02)00023-1.

Imai A, Sun X, Nishimura E, Papadimitriou S, 2005 *Berth allocation in a container port: using a continuous location space approach. Transportation Research Part B-methodological* 39(3):199–221, URL http://dx.doi.org/10.1016/j.trb.2004.04.004.

IMO, 2018 *Initial IMO strategy on reduction of GHG emissions from ships.* Technical Report MEPC.304(72), International Maritime Organization, URL http://www.imo.org/en/OurWork/Environment/PollutionPrevention/AirPollution/Pages/GHG-Emissions.aspx, (Accessed on 04.05.2020).

IMO, 2020 *Reduction of GHG emissions from ships. Fourth IMO GHG study 2020.* Technical Report MEPC.304(72), International Maritime Organization, URL https://docs.imo.org/Documents/Detail.aspx?did=125134, (Accessed on 22.08.2020).

Iris C, Pacino D, Røpke S, 2017 *Improved formulations and an adaptive large neighborhood search heuristic for the integrated berth allocation and quay crane assignment problem. Transportation Research. Part E: Logistics and Transportation Review* 105:123–147, URL http://dx.doi.org/10.1016/j.tre.2017.06.013.

Iris C, Pacino D, Røpke S, Larsen A, 2015 *Integrated berth allocation and quay crane assignment problem: Set partitioning models and computational results. Transportation Research. Part E: Logistics and Transportation Review* 81:75–97, URL http://dx.doi.org/10.1016/j.tre.2015.06.008.

Jans R, 2010 *Classification of dantzig-wolfe reformulations for binary mixed integer programming problems. European Journal of Operational Research* 204(2):251–254, URL http://dx.doi.org/10.1016/j.ejor.2009.11.014.

Kontovas C, Psaraftis, 2011 *Reduction of emissions along the maritime inter modal container chain: Operational models and policies. Maritime Policy and Management* 38(4):451–469, URL http://dx.doi.org/10.1080/03088839.2011.588262.

Kordić S, Davidović T, Kovač N, Dragović B, 2016 *Combinatorial approach to exactly solving discrete and hybrid berth allocation problem. Applied Mathematical Modelling* 40(21-22):8952–8973, URL http://dx.doi.org/10.1016/j.apm.2016.05.004.

Krajewska MA, Kopfer H, Laporte G, Ropke S, Zaccour G, 2008 *Horizontal cooperation among freight carriers: request allocation and profit sharing. Journal of the Operational Research Society* 59(11):1483–1491, URL http://dx.doi.org/10.1057/palgrave.jors.2602489.

Kramer A, Lalla-Ruiz E, Iori M, Voß S, 2019 *Novel formulations and modeling enhancements for the dynamic berth allocation problem. European Journal of Operational Research* 278(1):170–185, URL http://dx.doi.org/10.1016/j.ejor.2019.03.036.

Lalla-Ruiz E, Expósito-Izquierdo C, Melián-Batista B, Moreno-Vega JM, 2016 *A set-partitioning-based model for the berth allocation problem under time-dependent limitations. European Journal of Operational Research* 250(3):1001–1012, URL http://dx.doi.org/10.1016/j.ejor.2015.10.021.

Lalla-Ruiz E, Melián-Batista B, Moreno-Vega JM, 2016 *A cooperative search for berth scheduling. Knowledge Engineering Review* 31(05):498–507, URL http://dx.doi.org/10.1017/s0269888916000266.

Lim A, 1998 *The berth planning problem. Operations Research Letters* 22(2-3):105–110, URL `http://dx.doi.org/10.1016/S0167-6377(98)00010-8`.

Maersk, 2021 *Maersk.* `https://www.maersk.com/`, accessed: 2021-09-15.

Magnanti TL, Orlin JB, Ahuja RK, 1993 *Network flows : theory, algorithms, and applications* (Prentice-Hall), ISBN 013617549x, 9780136175490.

Meisel F, Bierwirth C, 2009 *Heuristics for the integration of crane productivity in the berth allocation problem. Transportation Research Part E: Logistics and Transportation Review* 45(1):196–209, URL `http://dx.doi.org/10.1016/j.tre.2008.03.001`.

Meng Q, Wang S, 2011 *Optimal operating strategy for a long-haul liner service route. European Journal of Operational Research* 215(1):105–114, URL `http://dx.doi.org/10.1016/j.ejor.2011.05.057,10.1016/j.ejor.2011.05.057`.

Navis, 2021 *Navis.* `https://www.navis.com/`, accessed: 2021-09-07.

Notteboom TE, Parola F, Satta G, Pallis AA, 2017 *The relationship between port choice and terminal involvement of alliance members in container shipping. Journal of Transport Geography* 64:158–173, URL `http://dx.doi.org/10.1016/j.jtrangeo.2017.09.002`.

Notteboom TE, Vernimmen B, 2009 *The effect of high fuel costs on liner service configuration in container shipping. Journal of Transport Geography* 17(5):325–337, URL `http://dx.doi.org/10.1016/j.jtrangeo.2008.05.003`.

Özener OO, Ergun O, Savelsbergh M, 2011 *Lane-exchange mechanisms for truckload carrier collaboration. Transportation Science* 45(1):1–17, URL `http://dx.doi.org/10.1287/trsc.1100.0327`.

Packard W, 1995 *Shipping Pools.* Business of Shipping S (Lloyd's of London Press), ISBN 9781850445128, URL `https://books.google.dk/books?id=g7ArAQAAIAAJ`.

Pang KW, Liu J, 2014 *An integrated model for ship routing with transshipment and berth allocation. Iie Transactions (institute of Industrial Engineers)* 46(12):1357–1370, URL `http://dx.doi.org/10.1080/0740817X.2014.889334`.

Portchain, 2021 `https://www.portchain.com`, accessed: 2021-09-06.

Psaraftis HN, Kontovas CA, 2013 *Speed models for energy-efficient maritime transportation: A taxonomy and survey. Transportation Research Part C-emerging Technologies* 26:331–351, URL `http://dx.doi.org/10.1016/j.trc.2012.09.012`.

Psaraftis HN, Kontovas CA, 2015a *Green maritime transportation: Speed and route optimization. International Series in Operations Research and Management Science* 226:299–349, URL `http://dx.doi.org/10.1007/978-3-319-17175-3_9`.

Psaraftis HN, Kontovas CA, 2015b *Slow steaming in maritime transportation: Fundamentals, trade-offs, and decision models. Handbook of Ocean Container Transport Logistics* 315–358, URL `http://dx.doi.org/10.1007/978-3-319-11891-8_11`.

Pujats K, Golias M, Konur D, 2020 *A review of game theory applications for seaport cooperation and competition. Journal of Marine Science and Engineering* 8(2):100, URL http://dx.doi.org/10.3390/jmse8020100.

RBS, 2021 *Realtime Business Solutions.* https://www.rbs-tops.com/, accessed: 2021-09-06.

Reinhardt LB, Plum CE, Pisinger D, Sigurd MM, Vial GT, 2016 *The liner shipping berth scheduling problem with transit times. Transportation Research. Part E: Logistics and Transportation Review* 86:116–128, URL http://dx.doi.org/10.1016/j.tre.2015.12.006.

Saadaoui Y, Umang N, Frejinger E, 2015 *A column generation framework for berth scheduling at port terminals* (CIRRELT, Centre interuniversitaire de recherche sur les réseaux d'entreprise . . .).

Saeed N, Larsen OI, 2010 *An application of cooperative game among container terminals of one port. European Journal of Operational Research* 203(2):393–403, URL http://dx.doi.org/10.1016/j.ejor.2009.07.019.

Sealytix, 2021 *Sealytix.* https://www.sealytix.com/, accessed: 2021-09-20.

Shapley LS, 1953 *A value for n-person games. Contributions to the Theory of Games* 2(28):307–317.

Ship & Bunker, 2021 *Global Average Bunker Price - VLSFO.* https://shipandbunker.com/prices/av/global/av-glb-global-average-bunker-price#VLSFO, accessed: 2021-03-21.

Song DW, 2003 *Port co-opetition in concept and practice. Maritime Policy and Management* 30(1):29–44, URL http://dx.doi.org/10.1080/0308883032000051612.

Song DW, Panayides PM, 2002 *A conceptual application of cooperative game theory to liner shipping strategic alliances. Maritime Policy and Management* 29(3):285–301, URL http://dx.doi.org/10.1080/03088830210132632.

Steenken D, Voß S, Stahlbock R, 2004 *Container terminal operation and operations research - a classification and literature review. Or Spectrum* 26(1):3–49, URL http://dx.doi.org/10.1007/s00291-003-0157-z.

Sun B, Niu B, Xu H, Ying W, 2018 *Cooperative optimization for port and shipping line with unpredictable disturbance consideration. Icnc-fskd 2018 - 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* 8686901, 113–118, URL http://dx.doi.org/10.1109/FSKD.2018.8686901.

TGI, 2021 *TGI Maritime Software.* https://www.tgims.com/, accessed: 2021-09-06.

UNCTAD, 2019 *Review of Maritime Transport 2019.* URL http://dx.doi.org/https://doi.org/https://doi.org/10.18356/17932789-en, (Accessed on 27.06.2020).

Venturini G, Iris C, Kontovas CA, Larsen A, 2017 *The multi-port berth allocation problem with speed optimization and emission considerations. Transportation Research. Part D: Transport and Environment* 54:142–159, URL http://dx.doi.org/10.1016/j.trd.2017.05.002.

Martin-Iradi, Pacino, and Ropke: *The multi-port berth allocation problem with speed optimization*
Article submitted to ; manuscript no.

51

Wang H, Wang S, Meng Q, 2014 *Simultaneous optimization of schedule coordination and cargo allocation for liner container shipping networks. Transportation Research Part E: Logistics and Transportation Review* 70(1):261–273, URL `http://dx.doi.org/10.1016/j.tre.2014.07.005`.

Wang S, Liu Z, Qu X, 2015 *Collaborative mechanisms for berth allocation. Advanced Engineering Informatics* 29(3):572, 332–338, URL `http://dx.doi.org/10.1016/j.aei.2014.12.003`.

Wang S, Meng Q, 2012 *Liner ship route schedule design with sea contingency time and port time uncertainty. Transportation Research Part B* 46(5):615–633, URL `http://dx.doi.org/10.1016/j.trb.2012.01.003,10.1016/j.trb.2012.01.003`.

Wang S, Meng Q, Liu Z, 2013a *Bunker consumption optimization methods in shipping: A critical review and extensions. Transportation Research Part E: Logistics and Transportation Review* 53(1):49–62, URL `http://dx.doi.org/10.1016/j.tre.2013.02.003`.

Wang S, Meng Q, Liu Z, 2013b *A note on "berth allocation considering fuel consumption and vessel emissions". Transportation Research Part E: Logistics and Transportation Review* 49(1):48–54, URL `http://dx.doi.org/10.1016/j.tre.2012.07.002`.

Wen M, Larsen R, Røpke S, Petersen HL, Madsen OB, 2019 *Centralised horizontal cooperation and profit sharing in a shipping pool. Journal of the Operational Research Society* 70(5):737–750, URL `http://dx.doi.org/10.1080/01605682.2018.1457481`.

Wen M, Pacino D, Kontovas C, 2017 *A multiple ship routing and speed optimization problem under time, cost and environmental objectives. Transportation Research. Part D: Transport and Environment* 52(A):303–321, URL `http://dx.doi.org/10.1016/j.trd.2017.03.009`.

World Shipping Council, 2015 *Some observations on port congestion, vessel size and vessel sharing agreements*. Technical report, World Shipping Council, URL `https://www.worldshipping.org/industry-issues/transportation-infrastructure/Observations_on_Port_Congestion_Vessel_Size_and_VSA_May_28_2015.pdf`, (Accessed on 07.03.2021).

Zhen L, Hu Z, Yan R, Zhuge D, Wang S, 2020 *Route and speed optimization for liner ships under emission control policies. Transportation Research Part C: Emerging Technologies* 110:330–345, URL `http://dx.doi.org/10.1016/j.trc.2019.11.004`.