Computers & Industrial Engineering 124 (2018) 167–179

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Mathematical programming formulations for the strategic berth template problem

Çağatay Iris[a], Eduardo Lalla-Ruiz[b,c], Jasmine Siu Lee Lam[a,*], Stefan Voß[b,d]

[a] School of Civil and Environmental Engineering, Nanyang Technological University, 639798 Singapore, Singapore
[b] University of Hamburg, Institute of Information Systems (IWI), Von-Melle-Park 5, 20146 Hamburg, Germany
[c] Department of Industrial Engineering and Business Information Systems, Faculty of Behavioural Management and Social Sciences, University of Twente, 7522NB Enschede, The Netherlands
[d] Escuela de Ingeniería Industrial, Pontificia Universidad Católica de Valparaíso, Avenida Brasil 2241, Valparaíso 2362807, Chile

ARTICLE INFO

Keywords:
Maritime transportation
Berth allocation problem
Strategic planning
Set-packing formulation

ABSTRACT

The strategic berth template problem (SBTP) is an important strategic problem arising at the seaside of container terminals. It aims at supporting terminal managers in deciding which calling ships should be accepted, and it covers determining the most appropriate berth template for the accepted incoming traffic. In this work, we propose and evaluate two formulations. One is based on a conceptual but yet unexplored mathematical model and another is based on a generalized set-packing problem (GSP). Both formulations are assessed on a well-defined set of problem instances. The results indicate that the GSP-based optimization model exhibits a relevant performance, providing optimal solutions within reasonable time for most of the instances considered.

1. Introduction

The transportation of freight using containers is one of the main engines of the global economy because containerization allows to reduce transport costs while enabling rapid transfer between transportation modes. For countries and regions, the use of containers has an outstanding importance since it integrates different economies and establishes relations among them by means of global multi-modal supply chains. In this context, the United Nations Conference on Trade And Development (UNCTAD) publishes the Review of Maritime Transport (see United Nations Conference on Trade & Development, 2014) which indicates that the international seaborne trade has increased to 160 million twenty-foot equivalent units (TEUs) in 2014. Such a volume gives rise to congested scenarios where terminal managers have to provide efficient solutions to capture the majority of the incoming ship traffic. In this sense, a poor utilization of the main seaside resources, such as berths, badly affects other operations taking place at the yardside (e.g., container storage allocation, handling equipment management, container transshipment management) and landside (e.g., truck appointments).

The previous discussion leads to the definition of the berth allocation problem (BAP) which aims at determining the berthing position and berthing time for each ship arriving at the terminal within a given planning horizon. According to the planning level considered, three different variants of the BAP may arise:

- *Strategic*. The decisions covered at this level consider a time frame ranging from one year to several years. This longer time horizon includes decisions such as establishing shared and dedicated berths, arrangement of contracts and cooperation between terminals and shipping companies, etc. Works in this category are (Hendriks, Armbruster, Laumanns, Lefeber, & Udding, 2012; Imai, Yamakawa, & Huang, 2014).
- *Tactical*. The decisions arising at this level cover operations in a time horizon that ranges from one week up to several months. Some of the aspects that can be addressed at this level are the tactical yard templates, quay crane assignments with related working profiles, transshipment flows between ships, etc. The interested reader is referred to the following studies (Giallombardo, Moccia, Salani, & Vacca, 2010; Huang, Suprayogi, & Ariantini, 2016; Jin, Lee, & Hu, 2015; Lalla-Ruiz, González-Velarde, Melián-Batista, & Moreno-Vega, 2014; Moorthy & Teo, 2006; Zhen, Chew, & Lee, 2011).
- *Operational*. This level considers the shortest time horizon among the three planning levels. It covers decisions ranging from one up to several days. Such problems are mostly aimed at minimizing the berthing and departure delays of container ships and/or idle times

* Corresponding author.
E-mail addresses: ciris@ntu.edu.sg (Ç. Iris), eduardo.lalla-ruiz@uni-hamburg.de, e.a.lalla@utwente.nl (E. Lalla-Ruiz), sllam@ntu.edu.sg (J.S.L. Lam), stefan.voss@uni-hamburg.de (S. Voß).

https://doi.org/10.1016/j.cie.2018.07.003
Received 13 November 2017; Received in revised form 8 May 2018; Accepted 3 July 2018
Available online 07 July 2018
0360-8352/

of berths. Some studies at this level are in Imai, Nishimura, and Papadimitriou (2001), Cordeau, Laporte, Legato, and Moccia (2005), Monaco and Sammarra (2007), Hoffarth and Voß (1994), Lalla-Ruiz, Melián-Batista, and Moreno-Vega (2012), Hu (2015),Iris, Pacino, Ropke, and Larsen (2015), Ribeiro, Mauri, de Castro Beluco, Lorena, and Laporte (2016), and Venturini, Iris, Kontovas, and Larsen (2017).

In the related literature, the majority of the papers study the BAP and several variants at the operational level. To a lesser extent, some papers address the BAP at the tactical level and only a few of them study the BAP at the strategic level. Recently, Imai et al. (2014) proposed the strategic berth template problem (SBTP) which deals with relatively long-term decisions regarding berth planning. In their paper, the berth template is defined as a set of berth-windows of selected calling ships during a fixed-length prototype planning horizon. The long-term calling ship profiles (ship sizes, processing times, etc.) are obtained from shipping lines. Liner shipping companies usually call each port on the same day of every week (Wang, Meng, & Lee, 2016; Wang, Liu, & Qu, 2017). Wang et al. (2016) noted that helping the terminals for a better berth template is one of the main reasons for providing weekly schedules. Commonly, different ships call the terminal each week, but the sizes of the ships are quite similar for each week. Terminals determine longer term berth templates that consider a single prototype week which is very similar in the following weeks. The berthing capacity in one week is a limited resource, and candidate mega ships and their feeders might require a total berthing time which is beyond the capacity. In that case, berth templates also consider the strategic decision to reject some calls and optimize the use of the berthing area for the calls to be served. This allows to address congested scenarios or those cases in which the terminal has limited resources. Such templates have a long-term aspect. They assume that the processing time of a ship is roughly known, and it does not differ between the assigned berths and the weeks of the year.

In this paper, the strategic berth template problem is addressed from a modeling perspective. It comprises the discussion concerning the initial formulation which was theoretically proposed in Imai et al. (2014) but remained computationally unexplored as yet. In this regard, by fully considering the SBTP as presented in Imai et al. (2014) we propose a generalized set-packing problem (GSP) formulation to efficiently solve it. Both formulations are also improved through the inclusion of additional lower bounds. For assessing the performance of the mathematical formulations and the lower bounds, a well-defined set of problem instances is proposed. Extensive computational experiments show that the GSP formulation is able to solve reasonably sized instances in time limits deemed practical and it clearly outperforms the other in terms of linear bounds' quality and computational times. The results also show a meaningful contribution of the incorporated lower bounds on the solvability of both optimization models. Finally, the composition of selected calls is evaluated with respect to different ship types and terminal scales.

The outline of this paper is as follows. In Section 2, the works related to the SBTP are discussed. The strategic berth template problem is presented in Section 3. Later, the mathematical models are discussed in Section 4. Extensive computational experiments are reported in Section 5. Finally, conclusions and further research opportunities are presented in Section 6.

2. Literature review

The berth allocation problem (BAP) aims at assigning a berthing position and time to each incoming ship at a container terminal. It has been greatly addressed in the literature producing several variants (see Bierwirth & Meisel, 2010, 2015). The variant of the BAP in this work considers a dynamic arrival of ships (*i.e.* they arrive along the time horizon and the arrival time bounds the berthing start time), and ships have to be allocated within a discrete quay (*i.e.* the quay is divided into separate sections termed as berths and each ship fits into one berth). Usually, this problem variant is referred to as the dynamic BAP. In this section, we first review those works directly related to that variant of the BAP at the operational level and later those studies related to the strategic planning level.

The dynamic (and discrete) berth allocation problem (DBAP) defined within an operational level was proposed by Imai et al. (2001) and later extended for including ship priorities in Imai, Nishimura, and Papadimitriou (2003). Monaco and Sammarra (2007) dealt with the DBAP presenting a new formulation, and they proved that their proposed formulation is stronger than that of Nishimura, Imai, and Papadimitriou (2001) and Imai et al. (2001). They noted the importance of having stronger formulations that can be used for solving instances to optimality. Additionally, they presented an efficient heuristic based on Lagrangean decomposition. Cordeau et al. (2005) proposed a formulation for the DBAP based on the multi-depot vehicle routing problem with time windows (MD-VRPTW). The authors studied the problem for two variations in terms of quay layout, namely, discrete and hybrid (*i.e.* the quay is divided into berths of different length and a vessel can occupy more than one berth). In addition, the authors proposed the inclusion of time windows for ships and berths. In order to solve this problem, two tabu search heuristics were presented.

Buhrkal, Zuglian, Ropke, Larsen, and Lusby (2011) reviewed the three well-known mathematical models for the operational DBAP. They indicated that the most proper one in terms of solution quality and solving time is the one based on a generalized set-partitioning problem (GSPP) proposed by Christensen and Holst (2008). Lalla-Ruiz et al. (2012) proposed a tabu search algorithm with path-relinking to solve the discrete DBAP. In that work the authors reported that for some problem instances the formulation provided in Christensen and Holst (2008) could not be solved due to a memory fault status. In such cases, heuristic methods can be used to solve this problem. Lalla-Ruiz and Voß (2016) developed the first matheuristic approach for this problem, *i.e.* POPMUSIC, and tested it on the instances proposed in Cordeau et al. (2005, 2012). The method allowed to treat the instances by means of the GSPP formulation by decomposing them into smaller and computationally tractable subproblems. The computational results reflect a competitive performance, but it requires quite a considerable amount of computational time.

A recent variation, regarding the DBAP, includes the consideration of tides. Xu, Li, and Leung (2012) addressed the BAP under time-dependent limitations (BAPTL) which refers to tidal conditions. In that work, the water depth of the berths changes during the planning horizon for different tide periods and there are some ships which are sensitive to the depth at the berth. In Xu et al. (2012) the BAPTL was formulated as a mixed integer linear program and it was solved with a number of heuristics. Lalla-Ruiz, Voß, Expósito-Izquierdo, Melián-Batista, and Moreno-Vega (2017) proposed a POPMUSIC approach for the BAPTL. Different to Lalla-Ruiz and Voß (2016), they considered a novel way of building the sub-problems while decomposing the problem instances. In addition, some reinforcement constraints as well as the extension to include time windows for the BAPTL were studied in Lalla-Ruiz, Expósito-Izquierdo, Melián-Batista, and Moreno-Vega (2016).

Concerning the strategic level, there are few recent papers that address the dynamic berth allocation problem. Hendriks et al. (2012) considered the strategic allocation of cyclically calling ships at multiple terminals within the same port. In that work they aimed at reducing the imbalance concerning the quay crane workload as well as minimizing the amount of inter-terminal container transport between the terminals. The authors considered, as proposed by Moorthy and Teo (2006), a cyclic time horizon. If a ship arrives at the end of a period (cycle), it might depart at the beginning of the next cycle. A mixed-integer linear programming formulation and a two-step optimization approach were investigated. Imai et al. (2014) proposed the strategic berth template

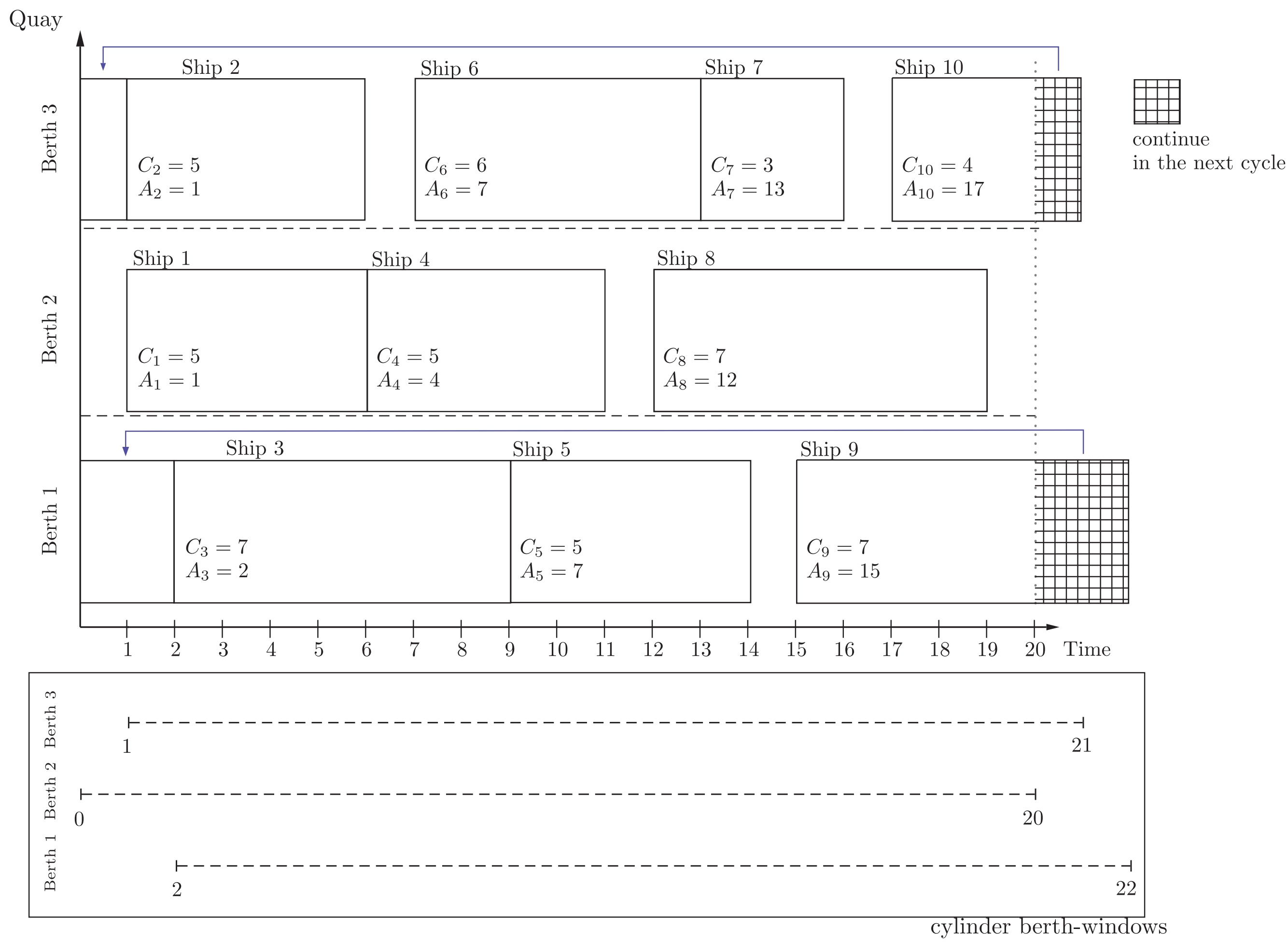

Fig. 1. Example of a solution for the SBTP considering 3 berths and 10 ships.

problem (SBTP). The problem is defined on a strategic planning level and includes decisions regarding the selection of ship calls to be served and the assignment of proper berth-windows for selected ships within a cylinder horizon (*i.e.* a cyclically repeating planning horizon). Furthermore, the simultaneous treatment of mother and feeder ships under the hub-and-spoke operation is considered in the SBTP. The authors proposed a mathematical formulation based on that of Imai et al. (2001) and three Lagrangian heuristics, each with three different weighted handling times for the subgradient procedure. The computational results indicate that the problem is hardly solvable by means of the given Lagrangian methods. Jin et al. (2015) addressed the quayside berthing congestion from a tactical planning viewpoint by balancing the workload distribution over time. In doing so, the authors considered the berth template design problem together with the yard template design problem with the goal of minimizing spatial-related costs due to container flows and temporal costs regarding quay-side workload imbalance. To solve this problem, the authors proposed a set-covering problem formulation and two variants of column generation methods. Recently, Huang et al. (2016) addressed the berth template problem in a continuous quay as a mid-term tactical decision problem. The authors extended a formulation from Kim and Moon (2003) by including constraints regarding the nature of the cyclic planning horizon. Additionally, the authors proposed two heuristic approaches.

The SBTP in this work finds similarities with the unrelated parallel machine scheduling problem (see Mokotoff, 2001) in which n jobs (ships) are scheduled on m parallel unrelated machines (berths). Some special properties of the SBTP such as ship call rejections, binding ship arrival times, cyclic ship calls are addressed in the parallel machine scheduling literature. Lu, Zhang, and Yuan (2008) studied an unbounded parallel machine scheduling problem with release dates and job rejection costs. They proved that the problem is binary $\mathcal{NP}$-hard and proposed approximation schemes to solve the problem. Zhang and Lu (2016) studied an on-line version of a similar problem in which jobs arrive in an arbitrary fashion. They provided a pseudo-polynomial-time algorithm and a fully polynomial-time approximation scheme if the number of machines is known. Gerstl and Mosheiov (2012) studied the parallel machine scheduling problem with job rejections and position-dependent processing times. The authors suggested efficient algorithms that require a solution of a set of linear assignment problems (relaxed version of the problem). Shirvani, Ruiz, and Shadrokh (2014) addressed the cyclic scheduling of jobs in parallel machines with release dates and due dates. They proposed a mathematical model and heuristic algorithm to solve the cyclic jobs variant of the problem. For a more general version of the parallel machine scheduling problem with job release dates, Yalaoui and Chu (2006) applied an exact branch-and-bound approach. Lower bounding methods have also been discussed in that paper by relaxing release dates and preemption rules.

3. Problem description

The strategic berth template problem (SBTP) in Imai et al. (2014) addresses the berth allocation problem at a strategic level considering the selection of calling ships in a cyclic time horizon. The SBTP considers a set of cyclically calling ships (V) from which a subset $S_v \subseteq V$ has to be selected for being serviced. Those selected calling ships have to be berthed within a quay partitioned into identical berths. Thus, for each selected ship $j \in V$, a berthing position $i \in B$ and berthing time window have to be determined. Since most container ships call at a terminal on a weekly basis, the SBTP considers a cylinder with a length H. That is, the SBTP utilizes a fixed planning horizon (related to cylinder) which is repeated every H time steps as done by Moorthy and Teo (2006). Although the cylinder length remains the same, it does not necessarily start at the first time unit. The cylinder for a berth at the end of its length can be extended as long as the first ship starts berthing for that berth. In other words, we are allowed to shift the cylinder as long as there is available time in the start of the planning horizon. The objective of the SBTP is to find the best selection of calling ships by minimizing delays in the berthing time and rejected berthing hours. A delay in a ship berthing time corresponds to the waiting time of the ship until it berths once it has already arrived at the terminal. The rejected berthing hours in the objective are proportional to the processing time of the rejected ship call.

An example of a solution for the SBTP is illustrated in Fig. 1. The SBTP leans itself into a two-dimensional space definition; one dimension is spatial where each ship occupies one berth, and the other dimension is temporal where each ship arrives in a cyclic manner and the arrival time of the ship sets a lower bound on the berthing time. This example considers the case where 10 ships have to be berthed in 3 berths along with a cyclic time horizon of $H = 20$. Each ship $j \in V$ is represented by a rectangle that points out the berthing time window and assigned berth. Inside each rectangle, the corresponding arrival time (A_j) and processing time (C_j) for each ship are reported. As can be checked in the example, some ships (e.g. ships 2 and 6) can be allocated within the time horizon. Nevertheless, other ships such as 9 and 10 start to be served at the end of a cycle, so likely in the next cycle they continue to be handled.

Furthermore, the planning horizon can be shifted for different berths thanks to the consideration of it as a cylinder. Berths 1 and 3 have a shifted cylinder. That is, for berth 3 in Fig. 1, ship 10 starts berthing at time unit 17 and finishes operations at time unit 21. Note that the cylinder length is 20 time units. The template is still feasible since the first ship being served at berth 3 is ship 2 and it starts berthing at time unit 1. This means the cylinder is shifted and finishes at time unit 21 ($=20 + 1$). Summing up, the cylinder of berth 3 is between period 1 and 21. Berth 3 will still be occupied by ship 10 from the previous week to the current between time units 0 and 1, but it is not the ship in the prototype week.

Depending on the size and dependency of the ships, in the SBTP two types of ships are considered: *(i)* mother ships (*i.e.* large-sized ships) and *(ii)* feeder ships (*i.e.* small-sized ships). In many terminals, some mother and feeder ships have strong transhipment relations where containers brought by a mother ship must be transhipped to feeder ships, and vice versa. These ships are contractually attached to each other. Since the SBTP allows a ship call rejection, these relationships should be considered. Assume that a mother ship is rejected, there are some feeder ships which are strongly linked to the mother ship with contracts. It does not make sense for these feeders to call the port since their mother ship was mainly supposed to supply their cargo. For this reason, we also consider sets of linked ships. All ships in each set either call the port or they are all rejected. We now define a single set R_{jj^1}, and it holds a zero or one value for each ship pair (j, j^1). If two ships j and j^1 are supposed to be linked, R_{jj^1} takes the value of one, otherwise, it is zero. And, it is ensured that $R_{jj^1} = R_{j^1 j}$. This problem can also handle multiple calls per week (biweekly, daily, etc.) by one service (*i.e.* liner route). In that case, it is required that the R_{jj^1} value for two ships (j, j^1) of the same service should be one.

The SBTP differs from the traditional DBAP in various aspects. The SBTP conceptualizes a prototype week for the quay where the calling ships' characteristics are roughly estimated, while the DBAP mostly deals with more precise information due to its operational planning horizon. Because the SBTP considers cyclic properties of the ship calls (can be weekly, bi-weekly, etc.), the determination of the planning horizon (*i.e.* cylinder starting and ending time for each berth) is also a part of the decision making. The option to reject some ship calls is another major difference between the SBTP and the DBAP. The SBTP allows terminals to reject some calls as it is a strategic problem with the flexibility of managing the contracts and conducting negotiations with shipping companies. This contrasts with the DBAP where the planning horizon is mostly fixed and the planners schedule all contracted ships in the given planning horizon. Another strategic aspect linked to the SBTP is the consideration of relationships among mother and feeder ships, as well as the selection of calling ships. At this point, it can be pointed out that the relationship between mother and feeder ships is not considered in the traditional DBAP.

4. Mathematical models and properties

In this section two formulations and various properties for the SBTP are presented.

4.1. Three-index formulation

The SBTP is presented with the aim of finding the best subset of calling ships to be served and their berthing order in their assigned berth. An optimization model is proposed to classify the problem, though it is not evaluated computationally. It seeks to determine an optimal plan for the best berthing position, arrival, and departure time of each selected ship. A solution also determines the berthing cycle (cylinder) which does not have to be the same for each berth. The SBTP has been formulated with the objective of minimizing the total time of delaying to berth all ships and the total time of rejected ship calls. We now detail the three-index formulation theoretically presented in Imai et al. (2014) followed by the enhancements composing the extended formulation.

There are three types of variables for this formulation. The binary variable x_{ijk} is one if ship $j = \{1, 2, ..., N\} \in V$ is berthed at berth $i \in B = \{1, 2, ..., S\}$ as the $k^{th} \in U = \{1, 2, ..., N\}$ ship, otherwise it is zero. The berth set B can form a union with a dummy berth which is indexed as berth 0 (such that $B \cup \{0\}$). This dummy berth corresponds to the rejection of a ship call,[1] *i.e.* all rejected ships are allocated to this berth. Thus, for instance, $x_{0,jk}$ takes the value of one for any ship j if the ship j is rejected to berth at the port. The variable y_{ijk} is an integer variable corresponding to the idle time of berth i between the departure of the $(k + 1)^{th}$ ship and the arrival of the k^{th} ship when ship j is served as the k^{th}. Here we should note that the service ordering indices ($k \in U = \{1, 2, ..., N\}$) take values in descending order of berthing. Let us explain the service order numbering with an example. If nine different ships are assigned to berth 1 and ship 4 is physically the first ship to be served at that berth, then the corresponding decision variable x_{149} takes the value of one, where as can be seen the value of k equals 9. Moreover, if the following ship is 3, then x_{138} takes the value of one, where in this case k equals 8, and so on. The same service ordering approach is followed for the y_{ijk} variables with respect to k indices. Note that for a service order k at berth i, the idle time y_{ijk} is the waiting time after the departure of the preceding ship. Such a service ordering was designed in order to formulate a traceable objective function (see further explanation below).

A pre-defined subset $P_k(\subset U)$ refers to a set of service orderings which are larger than k ($P_k = \{p | p > k \in U\}$). In other words, the subset P_k contains service orders which are before service order k. Finally, the integer variable Y_i refers to the opening time of the cycle for berth i. The parameters A_j, C_j correspond to the arrival and processing time for ship j, respectively. H is the length of the cylinder (the cycle time of the

[1] Note that the idea of a dummy berth was already proposed in Hoffarth and Voß (1994).

planning horizon), while g is the penalty time of rejecting a ship call. The model of the SBTP is as follows:

$$\min \sum_{i\in B}\sum_{j\in V}\sum_{k\in U}\{(k-1)C_j - A_j\}x_{ijk} + \sum_{i\in B}\sum_{j\in V}\sum_{k\in U} ky_{ijk} + \sum_{j\in V}\sum_{k\in U} gC_j x_{0,jk} \quad (1)$$

subject to

$$\sum_{i\in B\cup\{0\}}\sum_{k\in U} x_{ijk} = 1 \quad \forall\, j\in V \quad (2)$$

$$\sum_{j\in V} x_{ijk} \le 1 \quad \forall\, i\in B\cup\{0\}, \forall\, k\in U \quad (3)$$

$$\sum_{l\in V}\sum_{m\in P_k}(C_l x_{ilm} + y_{ilm}) + y_{ijk} - A_j x_{ijk} \geqslant 0 \quad \forall\, i\in B, \forall\, j\in V, \forall\, k\in U \quad (4)$$

$$\sum_{j\in V}\sum_{k\in U}(C_j x_{ijk} + y_{ijk}) - Y_i \leqslant H \quad \forall\, i\in B \quad (5)$$

$$\left(1-\sum_{j\in V} x_{ijk}\right)H + \sum_{l\in V}\sum_{m\in P_k}(C_l x_{ilm} + y_{ilm}) + \sum_{j\in V} y_{ijk} \geqslant Y_i \quad \forall\, i\in B, \forall\, k\in U \quad (6)$$

$$R_{jj^1}\sum_{i\in B}\sum_{k\in U} x_{ijk} = R_{j^1 j}\sum_{i\in B}\sum_{k\in U} x_{ij^1k} \quad \forall\, j, j^1(>j)\in V \quad (7)$$

$$Y_i \leqslant H \quad \forall\, i\in B \quad (8)$$

$$x_{ijk}\in\{0,1\} \quad \forall\, i\in B\cup\{0\}, \forall\, j\in V, \forall\, k\in U \quad (9)$$

$$0 \leqslant y_{ijk} \leqslant A_j \quad \forall\, i\in B, \forall\, j\in V, \forall\, k\in U \quad (10)$$

The objective function (1) corresponds to the delay in the berthing time (*i.e.* the difference between arrival time and berthing time) and time reflection of rejected ship calls. Since the above three-index formulation does not track the exact berthing time as a variable, the objective function is not formulated in an accumulative way of berthing times. Instead, a basic observation is used to derive the objective function. That is, the berthing time of each ship inherits the berth idle time and processing time of its predecessors in the same berth. The term $\sum_{i\in B}\sum_{j\in V}\sum_{k\in U}\{(k-1)C_j x_{ijk} + ky_{ijk}\}$ corresponds to the total accumulation of processing times and idle times of all ships in all service orders considering predecessors of each ship in each berthing order. In (1), $\sum_{i\in B}\sum_{j\in V}\sum_{k\in U} ky_{ijk}$ is the sum of berth idle times for all ships in all service orders considering predecessors and $\sum_{i\in B}\sum_{j\in V}\sum_{k\in U}(k-1)C_j x_{ijk}$ is the total processing time for all predecessors of all ships. When we subtract the sum of arrival times of all ships ($\sum_{i\in B}\sum_{j\in V}\sum_{k\in U} A_j x_{ijk}$) from the sum of berthing times of all ships ($\sum_{i\in B}\sum_{j\in V}\sum_{k\in U}\{(k-1)C_j x_{ijk} + ky_{ijk}\}$), the total delay time in berthing (*i.e.* total ship berthing time delay) can be obtained. The third term in (1), $\sum_{j\in V}\sum_{k\in U} C_j x_{0,jk}$, refers to the total time of rejected berthing hours (ship calls) and one should multiply it with g to convert it to a penalty time factor. Constraints (2) ensure that each ship is either assigned to one of the berths in one of the service orders or it is rejected to call the port (observe that the berth index is summed in the set of $B\cup\{0\}$). Constraints (3) guarantee that each berth and service order combination is used by at most one ship at any time. Constraints (4) aim at linking the arrival time of a ship with its predecessors, and it ensures that each ship is berthed after its arrival. For each berth, ship and service order combination in constraints (4), the ship j is allowed to berth ($x_{ijk} = 1$) only if the total processing and idle times of all preceding ships in the same berth and idle time before it berths ($\sum_{l\in V}\sum_{m\in P_k}(C_l x_{ilm} + y_{ilm}) + y_{ijk}$) are beyond the arrival time of the ship (A_j). Constraints (5), (6) and (8) manage the cylinder, and they together ensure that all selected ships will be served within the defined cylinder. Constraints (5) guarantee that the sum of all processing and idle times for a berth is less than the cylinder length considering the opening time of the cylinder (Y_i) for that berth. Constraints (6) set the opening time of the cylinder. If no ship is assigned for berth i in any berthing order (*i.e.* $\sum_{j\in V} x_{ijk} = 0$), Y_i is at most H (see also (8)). If a ship is assigned for a berthing order k (*i.e.* $\sum_{j\in V} x_{ijk} = 1$), the start of the cylinder for berth i (Y_i) is at most the sum of total times (berthing and idle) of preceding ships in the same berth and the idle time before the ship berths ($\sum_{l\in V}\sum_{m\in P_k}(C_l x_{ilm} + y_{ilm}) + \sum_{j\in V} y_{ijk}$). Equalities (7) are intended to match mother and feeder ships due to contract requirements, either both are served or rejected. Constraints (8) set the upper bound on the opening time of a cylinder which can be opened at the end of the cycle (H) at the latest. In this way, the planning horizon could be at most $2H$ in length which is called H_{max} from this point on. The domains of variables are shown in (9) and (10). In an optimal solution, a ship does not wait to berth as long as at least one berth is available and the ship has arrived at the port. This is due to the fact that the objective function is increasing by one unit with each delayed time unit and no contradiction is obtained. Constraints (10) allow that the idle time for ship j can be at most the arrival time of ship j. This upper bound is ensured in the way that the ship j would be the first ship to be serviced in that berth so that the berth should stay idle until A_j. The above model (1)–(10) is called SBTP in the remainder of the paper.

For a similar formulation modeling, Monaco and Sammarra (2007) proved that the idle time variable at a berth (y_{ijk}) is not linked to that specific ship which is berthed. Based on that they showed that the traditional formulation of Imai et al. (2001) can be strengthened by replacing y_{ijk} variables with y_{ik} where the ship subscript is dropped. Now we show that the formulation of Imai et al. (2014) for the SBTP inherits the same properties and a similar replacement can be made. Suppose that $\mathbb{C}^i_{k+1}$ corresponds to the completion time of the ship assigned to berth i at service order $k+1$, then the idle time of berth i at order k is $y_{ik} = \max\{0, \sum_{j\in V} A_j x_{ijk} - \mathbb{C}^i_{k+1}\}$. This property is based on the observation that a ship does not have to wait for berthing as long as its assigned berth is available, it is in accordance with the berthing order, and it has already arrived at the port.

We now use y_{ik} without loss of information, if $y_{ik} > 0$ in a feasible solution, then the corresponding ship for berth i in service order k cannot wait. We first replace constraints (4) with (11). The new constraints remain valid since for each berth and service order combination, there is only one selected ship, hence $\sum_{j\in V} A_j x_{ijk}$ will still correspond to the arrival time of a selected ship and is bounded by the idle time and processing time of the predecessors. Constraints (11) link the variables x_{ijk} and y_{ik}.

$$\sum_{m\in P_k}\left(y_{im} + \sum_{l\in V} C_l x_{ilm}\right) + y_{ik} - \sum_{j\in V} A_j x_{ijk} \geqslant 0 \quad \forall\, i\in B, \forall\, k\in U \quad (11)$$

The next two constraints refer to a reformulation of the cylinder, and constraints (5) and (6) are replaced with (12) and (13). The changes in the constraints are trivial, and they are due to the fact that these constraints set Y_i which depends on the specific berth.

$$\sum_{k\in U}\left(y_{ik} + \sum_{j\in V} C_j x_{ijk}\right) - Y_i \leqslant H \quad \forall\, i\in B \quad (12)$$

$$\left(1-\sum_{j\in V} x_{ijk}\right)H + \sum_{m\in P_k}\left(y_{im} + \sum_{l\in V} C_l x_{ilm}\right) + y_{ik} \geqslant Y_i \quad \forall\, i\in B, \forall\, k\in U \quad (13)$$

The domain (10) of the y_{jk} variable is replaced with constraints (14). They guarantee that the idle time in berth i for service order k should be at most the arrival time of the ship selected for berth i in service order k.

$$0 \leqslant y_{ik} \leqslant \underset{j\in V}{m} A_j x_{ijk} \quad \forall\, i\in B, \forall\, k\in U \quad (14)$$

Table 1
Structure of assignment matrix for given example.

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	RHS	
Vessel 1	1	1	1	1							≤ 1	a_{jz}
Vessel 2					1	1	1	1	1	1	≤ 1	
Berth1/Time1	1				1						≤ 1	b_{pz}
Berth1/Time2	1	1				1					≤ 1	
Berth1/Time3		1					1				≤ 1	
Berth2/Time1			1					1			≤ 1	
Berth2/Time2			1	1					1		≤ 1	
Berth2/Time3				1						1	≤ 1	

The objective function is rewritten using the new variables as in (15).

$$\min \sum_{i\in B}\sum_{j\in V}\sum_{k\in U}\{(k-1)C_j-A_j\}x_{ijk} + \sum_{i\in B}\sum_{k\in U} ky_{ik} + \sum_{j\in V}\sum_{k\in U} gC_j x_{0,jk} \tag{15}$$

Since all berths are identical, the solution space holds many symmetrical solutions. The formulation is indifferent between alternative berths so that we impose some symmetry breaking constraints. Constraints (16) arrange the berths in lexicographical order, and it breaks the symmetry. Jans (2009) showed the efficiency of similar constraints for some specific lot sizing problems on parallel machines.

$$\sum_{j\in V}\sum_{k\in U} x_{ijk} \leqslant \sum_{j\in V}\sum_{k\in U} x_{i+1,jk} \quad \forall\ i\in B\backslash\{S\} \tag{16}$$

The enhanced SBTP model with the suggested enhancements proposed in this section will be called SBTP+ in the remaining of the paper.

4.2. Generalized set-packing problem reformulation

In this section we formulate the SBTP as a generalized set-packing problem (GSP) model in which a variable represents one feasible assignment (or column) of a single ship call to a position (*i.e.* the combination of time and berth). The a priori generation of feasible assignments (or columns) should be first clarified. For each vessel j, the berth to moore can be any integer in the set $\{1, 2,\ldots, S\}$. For each berth in this set and ship j, the berthing time can be any integer between A_j and $H_{max}-C_j$. Columns for each ship are generated iteratively, the method goes through all alternatives (e.g. berth 1/time A_j, berth 1/time A_j+1, …, berth 1/time $H_{max}-C_j$, berth 2/time A_j, …, berth 2/time $H_{max}-C_j$, …, berth S/time A_j, …, berth S/time $H_{max}-C_j$) and generates berth and time combinations one by one.

The set of assignments is denoted by Ω. We define two matrices (a_{jz}), (b_{pz}) both containing $|\Omega|$ columns. Matrix (a_{jz}) contains a row for each ship call. Each element a_{jz} is binary and it is 1 if variable $z\in\Omega$ represents an assignment of ship $j\in V$. Binary matrix (b_{pz}) contains a row per position. The entry b_{pz} is one if position $p\in P$ is occupied in the assignment that variable z represents. The above mentioned assignment generation method constructs matrices which satisfy feasibility of all implicit constraints. Each assignment z has a delay value (d_z) with respect to the time of berthing. This lateness is calculated for each assignment with the information on the berthing time. In the model, there are S berths. Let P be the set of positions (berth, time combinations) that a ship call can occupy. The set P contains $2\cdot H\cdot S$ elements. The set T covers time periods in which a ship can call the port ($t\in T=\{1, 2, ..,H, ..,H_{max}\}$) where H_{max} is $2H$. The decision variable y_z, $z\in\Omega$ corresponds to each assignment, thus it takes the value of one if assignment z is part of the solution.

The GSP model is:

$$\min \sum_{z\in\Omega} d_z y_z + \sum_{j\in V} [gC_j(1-\sum_{z\in\Omega} a_{jz}y_z)] \tag{17}$$

subject to

$$\sum_{z\in\Omega} a_{jz}y_z \leq 1 \quad \forall\ j\in V \tag{18}$$

$$\sum_{z\in\Omega} b_{pz}y_z \leq 1 \quad \forall\ p\in P \tag{19}$$

$$\sum_{j\in V}\sum_{t_1\in\{T|t_1>t\}}\sum_{z\in\Omega^T(b,t_1,j)} y_z + \left(1-\sum_{j\in V}\sum_{z\in\Omega^T(b,t-H+1,j)} y_z\right)(t-H_{max})\leq 0 \quad \forall\ b\in B, \forall\ t\in\{T|t\geqslant H,\ t<H_{max}\} \tag{20}$$

$$R_{jj^1}\sum_{z\in\Omega} a_{jz}y_z = R_{j^1j}\sum_{z\in\Omega} a_{j^1z}y_z \quad \forall\ j, j^1(>j)\in V \tag{21}$$

$$y_z\in\{0, 1\} \quad \forall\ z\in\Omega \tag{22}$$

The objective function (17) is composed of two parts. The first part considers the delay time of selected assignments, while the second part is the time reflection of rejected ship calls. The term ($\sum_{z\in\Omega} a_{jz}y_z$) corresponds to one if the ship j is selected to berth at the terminal. Constraints (18) ensure that each ship call is served at most once. Constraints (19) guarantee that each position is occupied by at most one ship call. In constraints (20), $\Omega^T(b, t_1, j)$ corresponds to the set of assignments for ship j occupying berth b at time unit t_1. Constraints (20) allow to shift the cylinder for each berth ensuring a length of H for the cylinder. The shifting is only allowed at the end of the planning horizon (H). For each period after H, the constraints check the corresponding earliest berthing start for berth b. Until the first call is realized for a given berth b, one can extend the cylinder H as the slack amount in the starting window. Let us explain how constraints (20) work with an illustrative example. Assume that two ships will be assigned as in Table 1 with an H of 152. We show the structure of (20) for a berth b (say berth 2) and a time unit t larger than H (say 154). If a ship is assigned to berth 2 at time unit 3 ($t-H+1$) (*i.e.* $\sum_{j\in V}\sum_{z\in\Omega^T(b,t-H+1,j)} y_z=1$), no ships can be scheduled later than 155 (*i.e.* $\sum_{j\in V}\sum_{t_1\in\{T|t_1>t\}}\sum_{z\in\Omega^T(b,t_1,j)} y_z\leq 0$) in order to ensure a cylinder length of 152. Otherwise, $H_{max}-t$ binds the number of time units in which berth 2 is occupied after 155. Constraints (21) ensure that a couple of a mother ship and calling feeder in a transshipment contract are both served or neither of them is served. In order not to increase the symmetry in the formulation, j^1 is considered for ships which are indexed higher than j. Since we do not label mother and feeder ships, this formulation gives us more flexibility. Finally, the domain of y_z, $z\in\Omega$ is presented in constraints (22).

In Table 1, a small illustrative example of the model structure for two ships and two berths is shown. Ship 1 requires 2 time units and Ship 2 requires only 1 time unit. The first two rows in the table correspond to the ship which the column is referring to. The values in these two rows construct a_{jz} matrices. The next six rows indicate the six available time/berth combinations (positions) and show which position each column occupies. The values in these six rows construct b_{pz} matrices. There are ten possible assignments for the ships 1 and 2 while the last column on the right shows the right-hand-side (RHS) of the constraint which is formulated for each row.

4.3. Lower bounds on the SBTP

In order to enhance the performance of the optimization models, we now present a lower bound on the rejected berthing hours. This bound is applicable for both formulations, and it is based on observations from parallel machine scheduling problems. Note that assigning N ships to S berths is a variant of a parallel machine scheduling problem. The makespan is the maximum completion time of berthing for the last ship at each berth. We first observe that the cylinder property only shifts the planning window of the berths, it does not enlarge the available berthing hours which is $S\cdot H$.

Property 1. *If the total processing time of all ships ($\sum_{j\in V} C_j$) is larger than $S\cdot H$, there will be at least one rejected ship call.*

Proof. Assume that the complete berth allocation is achieved without any idle time in any berths, the total required berthing hours is then $\sum_{j\in V} C_j$. This constitutes a lower bound on the required berthing hours. If this lower bound is greater than the available berthing hours ($S\cdot H$), there should be at least one rejected ship call. □

Lemma 1. *In any optimal schedule for SBTP, there is no berth idle time between ships if all of them are available at time 0.*

Proof. Let π be an optimal schedule for SBTP. The total makespan of all berths is $\sum_{j\in V} C_j$ when all ships are available at time 0 in π. Assume that an idle time of γ is captured in one berth, the new total makespan is $\sum_{j\in V} C_j + \gamma$ and this increases the objective function by γ. □

Note that although the above lemma belongs to the static version of the berth allocation problem (*i.e.* all the ships are already at the port when the planning horizon starts), it is later considered in the generation of the lower bound for the SBTP. In this regard, a relationship between the lower bound on rejection and processing times can be deduced by using Property 1 and Lemma 1. We now present a new property for the SBTP.

Property 2. *Relaxing job arrival times (i.e. assuming all ships are available at time 0) and allowing preemption (i.e. processing time of a ship is allowed to be split) result in a lower bounding problem on the sum of makespan of all machines (berths).*

Proof. See Proposition 7 and 8 in Yalaoui and Chu (2006). □

For the SBTP, one can formulate a lower bound on the rejected berthing hours. This is done by obtaining the lower bound on the total makespan as Property 2 points out. Considering that all ships are available at time 0 and the bound on the planning horizon is H. The remaining ship hours unserved after H should be rejected as Property 1 suggests. That means one can split the processing time of a ship at the time unit H, and then the remaining time units of operations after H should be rejected. This comes with the observation that the total rejected berthing hours is proportional to how much the makespan goes beyond H.

Lemma 2. *The lower bound on rejected berthing hours is $\sum_{j\in V} C_j - S\cdot H$.*

Proof. The term $\sum_{j\in V} C_j$ is the lower bound on the sum of each berth completion time (considering that all ships are available at time 0). Meanwhile, one can assume that the ship operations are preempted at time H, and remaining ship hours unserved after H are rejected to call the port. Since splitting the handling operations constitutes a lower bound on the makespan, the total remaining berthing hours after H are $\sum_{j\in V} C_j - S\cdot H$, and this generates a lower bound on the rejected berthing hours. □

One can formulate the following inequalities by use of Lemma 2. Constraint (23) is added to the three-index formulation, while constraint (24) is added to the GSP model. The RHS is the calculated parameter for both constraints, while the total rejected berthing hours is $\sum_{j\in V}\sum_{k\in U} C_j x_{0,jk}$ for the three-index formulation and $\sum_{j\in V} C_j(1-\sum_{z\in\Omega} a_{jz}y_z)$ for the GSP formulation.

$$\sum_{j\in V}\sum_{k\in U} C_j x_{0,jk} \geqslant \sum_{j\in V} C_j - S\cdot H \tag{23}$$

$$\sum_{j\in V} C_j\left(1-\sum_{z\in\Omega} a_{jz}y_z\right) \geqslant \sum_{j\in V} C_j - S\cdot H \tag{24}$$

In the rest of the paper, the Generalized Set-Packing Problem formulation for the SBTP is termed as GSP, while the one including the valid inequalities is termed as GSP+.

5. Computational results

This section is devoted to present the computational results of the optimization models discussed for the SBTP. All tests are run on a computer equipped with an Intel i5 3.20 GHz and 16 GB of RAM. Moreover, all optimization models have been implemented by using CPLEX 12.6[2] with an execution time limit of 3 h for each problem instance and CPLEX set to all-default.

For the assessment of the models, the authors of Imai et al. (2014) provided us with an example scenario instance and based on its structure and the indications provided in their paper, we generated a set of problem instances. Additionally, in this work we study scenarios with different numbers of calling ships, *i.e.* 50, 70, 100, and 150; and different values for the number of berths, *i.e.* 4, 8 and 12. In the benchmark suite, there are two versions of the composition of calling ships in terms of workload (feeder (F), medium (M) or jumbo (J) ship). In one version, the number of calling ship types is equal (33.3% F, 33.3% M, 33.4% J), and in the other version, the composition is 60% F, 30% M, 10% J. It should be noted that medium and jumbo ships (or sometimes called megaships) are considered mother ships in the SBTP. We also differentiate the processing times for each ship according to the ship characteristics in the benchmark. There are mainly two types as average (A) or high (H). If the handling time is average for an instance, the feeder ships' handling times (in hours) are distributed with $Uniform(4, 8)$, while medium ships have $Uniform(6, 10)$ and jumbo ships have $Uniform(8, 12)$. If the handling time is high, the handling times for feeder, medium and jumbo ships are $Uniform(8, 10)$, $Uniform(10, 14)$, $Uniform(14, 22)$, respectively. Based on the combination of these different instance parameters we have produced 96 instances. Moreover, to ease the presentation of the results, we present the results divided into those coming from scenarios with 4 and those from 8 and 12 berths. The length of the planning horizon is 152 h which is a prototype week that will be repeated cyclically. The number of connections between mother ship and feeder ship depends on the instance size. Usually 10–20% of all ships are in a mother-feeder link. Finally, similarly to Imai et al. (2014), the lost income for rejecting a ship call (g) is 10,000 times as much as the handling time of that ship (penalty value).

5.1. Problem instances considering 4 berths

As done in Imai et al. (2014), we tackle scenarios considering terminal with 4 berths for 50, 70 and 100 calling ships. Note again that in the original paper, the mathematical formulation reported in Imai et al. (2014) was not evaluated. Hence, with the aim of evaluating our extended formulation based on it, that formulation is executed for the first time with a black-box integer programming (IP) solver.

Tables 2–4 show the computational performance of *(i)* the optimization model proposed in Imai et al. (2014), SBTP; *(ii)* our extended formulation of the SBTP that includes the addition of our enhancements and tighter lower bounds, SBTP+; *(iii)* our proposed generalized set-packing formulation, GSP; *(iv)* GSP enhanced with the addition of tighter lower bounds, GSP+. In all tables, we report the upper bound (UB), the lower bound (LB), the relative error (Gap%) provided by CPLEX, and the computational time measured in seconds (t(s.)). The time to generate the assignments (columns) is less than 10 s for each instance, and this time is included in t(s.).

From the computational results the following observations can be made:

- The performance of the SBTP model, as presented in Imai et al. (2014), is assessed for the first time and it reports negative lower

[2] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

Table 2
Computational results for the problem instances considering 50 ships and 4 berths.

N	S	id	SBTP				SBTP +				GSP				GSP +			
			UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)
50	4	1	712	−2089.90	393.53	10,800	71	0.00	100	10,800	11	11.00	0.00	5.20	11	11.00	0.00	5.60
		2	1213	−1945.10	260.35	10,800	125	0.00	100	10,800	13	13.00	0.00	5.90	13	13.00	0.00	6.20
		3	1760	−2126.10	220.80	10,800	105	0.00	100	10,800	16	16.00	0.00	5.70	16	16.00	0.00	6.10
		4	1285	−2447.50	290.47	10,800	298	0.00	100	10,800	14	14.00	0.00	5.20	14	14.00	0.00	5.60
		5	1608	−1614.20	200.39	10,800	112	0.00	100	10,800	22	22.00	0.00	7.50	22	22.00	0.00	6.10
		6	1936	−1948.90	200.67	10,800	81	0.00	100	10,800	23	23.00	0.00	5.90	23	23.00	0.00	6.30
		7	1492	−2025.50	235.76	10,800	301	0.00	100	10,800	62	62	0.00	6.10	62	62	0.00	6.40
		8	943	−1564.50	265.91	10,800	102	0.00	100	10,800	23	23.00	0.00	5.70	23	23.00	0.00	6.10
		9	321,553	−464.10	100.14	10,800	300,431	300000.00	0.14	10,800	300,270	285.10	99.91	10,800	300,172	300149.90	0.01	10,800
		10	296,858	−847.70	100.29	10,800	271,081	270001.00	0.40	10,800	270,329	268.20	99.90	10,800	270,236	270211.00	0.01	10,800
		11	372,213	−845.80	100.23	10,800	351,262	350001.70	0.36	10,800	351,053	488.10	99.86	10,800	351,012	350359.90	0.19	10,800
		12	142,722	−668.90	100.47	10,800	140,650	140000.00	0.46	10,800	140,349	198.60	99.86	10,800	140,286	140142.20	0.10	10,800
		13	1272	−931.90	173.26	10,800	833	0.00	100	10,800	291	259.40	10.86	10,800	291	259.60	10.79	10,800
		14	1158	−1300.50	212.31	10,800	533	0.00	100	10,800	231	213.90	7.40	10,800	231	213.90	7.40	10,800
		15	880	−600.40	168.23	10,800	419	0.00	100	10,800	217	180.60	16.77	10,800	216	180.60	16.39	10,800
		16	1033	−1241.50	220.18	10,800	256	0.00	100	10,800	107	107.00	0.00	13.30	107	107.00	0.00	13.60
Average:			71789.88	−1416.41	202.69	10,800	66666.25	66250.17	75.09	10,800	66439.44	136.56	27.16	4728.8	66420.94	66363.01	2.18	4728.88

Table 3
Computational results for the problem instances considering 70 ships and 4 berths.

N	S	id	SBTP				SBTP +				GSP				GSP +			
			UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)
70	4	17	1,258,883	−2372.70	100.19	10,800	667	0	100.00	10,800	243	204.50	15.84	10,800	243	205	15.84	10,800
		18	8463	−2082.80	124.61	10,800	425	0	100.00	10,800	56	56.00	0.00	11.30	56	56	0.00	13.10
		19	11,516	−2245.20	119.50	10,800	710	0	100.00	10,800	64	64.00	0.00	7.30	64	64	0.00	8.60
		20	78,272	−1284.80	101.64	10,800	289	0	100.00	10,800	104	104.00	0.00	13.40	104	104	0.00	16.10
		21	1,502,486	−1310.30	100.09	10,800	943	0	100.00	10,800	612	612.00	0.00	2082.50	612	612	0.00	2384.10
		22	2,461,744	−1373.80	100.06	10,800	7864	0	100.00	10,800	217	217.00	0.00	18.10	217	217	0.00	22.10
		23	1,787,247	−2030.20	100.11	10,800	728	0	100.00	10,800	331	331.00	0.00	20.60	331	331	0.00	24.50
		24	1,217,498	−1626.70	100.13	10,800	855	0	100.00	10,800	215	215.00	0.00	221.40	215	215	0.00	265.20
		25	2,526,981	1055.00	99.96	10,800	2,504,345	2,500,000	0.17	10,800	2,500,232	321133.60	87.16	10,800	2,500,147	2,500,127	0.00	10,800
		26	2,443,668	1058.70	99.96	10,800	2,425,177	2,420,000	0.21	10,800	2,420,211	249446.50	89.69	10,800	2,420,086	2,420,086	0.00	778.90
		27	2,901,818	1285.50	99.96	10,800	2,592,851	2,590,000	0.11	10,800	2,590,136	386007.10	85.10	10,800	2,590,087	2,590,087	0.00	693.50
		28	2,819,395	1132.70	99.96	10,800	2,749,693	2,740,000	0.35	10,800	2,740,447	617171.50	77.48	10,800	2,740,108	2,740,108	0.00	1050.20
		29	1,443,211	347.80	99.98	10,800	1,415,096	1,410,000	0.36	10,800	1,410,300	1547.20	99.89	10,800	1,410,156	1,410,112	0.00	10,800
		30	3,578,481	508.40	99.99	10,800	1,481,661	1,480,002	0.11	10,800	1,480,280	1539.50	99.90	10,800	1,480,100	1,480,100	0.00	6436.30
		31	1,349,191	270.40	99.98	10,800	1,301,887	1,290,000	0.91	10,800	1,290,231	1254.80	99.90	10,800	1,290,129	1,290,129	0.00	658.10
		32	1,987,640	588.80	99.97	10,800	1,290,803	1,290,034	0.06	10,800	1,290,487	1736.60	99.87	10,800	1,290,183	1,290,183	0.00	917.90
Average:			1711030.88	−504.95	102.88	10,800	985874.63	982502.26	50.14	10,800	982760.38	98852.52	47.18	6223.41	982677.38	982670.97	0.99	2854.29

Table 4
Computational results for the problem instances considering 100 ships and 4 berths.

N	S	id	SBTP+ UB	SBTP+ LB	SBTP+ Gap(%)	SBTP+ t(s.)	GSP UB	GSP LB	GSP Gap(%)	GSP t(s.)	GSP+ UB	GSP+ LB	GSP+ Gap(%)	GSP+ t(s.)
100	4	33	1,407,896	770054.5	45.30	10,800	770,361	1346.30	99.83	10,800	770,219	770,219	0.00	3776.10
		34	655,781	620,000	5.46	10,800	620,291	614.90	99.90	10,800	620,132	620,127	0.00	10,800
		35	681,324	640,000	6.07	10,800	640,379	959.10	99.85	10,800	640,180	640,180	0.00	3382.90
		36	710,494	700014.4	1.47	10,800	700,539	789.60	99.89	10,800	700,324	700224.5	0.01	10,800
		37	2,249,696	2,240,000	0.43	10,800	2,240,526	47531.50	97.88	10,800	2,240,086	2,240,086	0.00	773.10
		38	1,895,820	1,880,000	0.83	10,800	1,884,548	2861.30	99.85	10,800	1,880,088	1,880,088	0.00	2718.20
		39	1,853,388	1,840,000	0.72	10,800	1,840,615	2302.30	99.87	10,800	1,840,115	1,840,115	0.00	3346.80
		40	2,177,972	2,140,000	1.74	10,800	2,143,009	4972.40	99.77	10,800	2,140,145	2,140,145	0.00	3264.60
		41	5,769,299	5,760,000	0.16	10,800	5,760,615	3558161.60	38.23	10,800	5,760,042	5,760,042	0.00	842.60
		42	5,992,311	5,980,000	0.21	10,800	5,980,178	3830036.60	35.95	10,800	5,980,053	5,980,053	0.00	1029.90
		43	5,802,201	5,780,000	0.38	10,800	5,780,283	3592412.70	37.85	10,800	5,780,054	5,780,054	0.00	1432.50
		44	5,978,725	5,970,000	0.15	10,800	5,970,138	3806925.00	36.23	10,800	5,970,052	5,970,039	0.00	10,800
		45	4,684,709	4,670,000	0.31	10,800	4,670,418	2479265.20	46.92	10,800	4,670,064	4670051.80	0.00	10,800
		46	4,453,606	4,440,000	0.31	10,800	4,440,178	2253358.50	49.25	10,800	4,440,085	4,440,085	0.00	4010.60
		47	4,745,614	4,740,000	0.12	10,800	4,740,240	2581140.50	45.55	10,800	4,740,062	4,740,062	0.00	924.30
		48	4,559,835	4,530,000	0.65	10,800	4,530,131	2362116.80	47.86	10,800	4,530,052	4,530,052	0.00	1791.60
Average:			3351166.94	3293754.31	4.02	10,800	3294528.06	1532799.64	70.92	10,800	3293859.56	3293851.46	0.00	4405.83

Table 5
Mean values of average upper bounds ($\overline{UB}$), computational times ($\overline{t(s.)}$), accepted ship calls ($\overline{Acc.}$), and rejected ship calls ($\overline{Rej.}$) for instances considering 4 berths.

Set N	Set S	SBTP $\overline{UB}$	SBTP $\overline{t(s.)}$	SBTP $\overline{Acc.}$	SBTP $\overline{Rej.}$	SBTP+ $\overline{UB}$	SBTP+ $\overline{t(s.)}$	SBTP+ $\overline{Acc.}$	SBTP+ $\overline{Rej.}$	GSP $\overline{UB}$	GSP $\overline{t(s.)}$	GSP $\overline{Acc.}$	GSP $\overline{Rej.}$	GSP+ $\overline{UB}$	GSP+ $\overline{t(s.)}$	GSP+ $\overline{Acc.}$	GSP+ $\overline{Rej.}$
50	4	71789.88	10,800	49.44	0.56	66666.25	10,800	49.44	0.56	66439.44	4728.78	49.44	0.56	66420.94	2.18	49.50	0.50
70	4	1711030.88	10,800	52.69	17.31	985874.63	10,800	61.56	8.44	982760.38	6223.41	61.69	8.31	982677.38	2854.29	61.44	8.56
100	4	–	–	–	–	3351166.94	–	68.63	31.38	3294528.06	10,800	68.63	31.38	3293859.56	4405.83	67.31	32.69
Average		891410.38	10,800	51.06	8.94	1467902.60	10,800	59.88	13.46	1447909.29	7250.73	59.92	13.42	1447652.63	2420.76	59.42	13.92

bounds. The quality of those bounds, at the light of the accompanying results from the other models, leads to a poor evaluation of the quality of the UBs. This, therefore, points out an important drawback of that SBTP formulation.

- The overall performance of our enhanced formulation based on the SBTP clearly outperforms its former version. At this point, it should be noted that for some instances, it is able to reach very near to optimal solutions. Additionally, the enhancements enable the SBTP model to solve large-sized problem instances that cannot be solved by SBTP.
- Introducing a new formulation such as GSP is highly beneficial for this problem. Clearly, GSP provides better bounds and exhibits a shorter running time compared to SBTP and SBTP+. Moreover, the use of GSP with additional lower bounds (GSP+) gives a relevant advantage in terms of lower bounds, especially for the small-sized instances of 50 ships where the upper bounds of both, GSP and GSP+, are very similar and there are some rejected ship calls. In this set of instances, the importance of having better bounds, which results on a better evaluation of the solutions, can be highlighted.
- As the size of the instances increases, the improvement of GSP+ over GSP becomes more clear. In terms of optimal solutions, GSP+ is able to solve more instances to optimality compared to GSP. In this respect, it should be noted that in those cases where both formulations provide the optimal solution the computational times are similar.
- If all formulations obtain the same lower bound, it is observed that as the size of the instances increases, the quality of the solutions in terms of gap is improved. Among all alternatives, GSP+ is the best modeling option.

In Table 5, each set of the problem instances is compared according

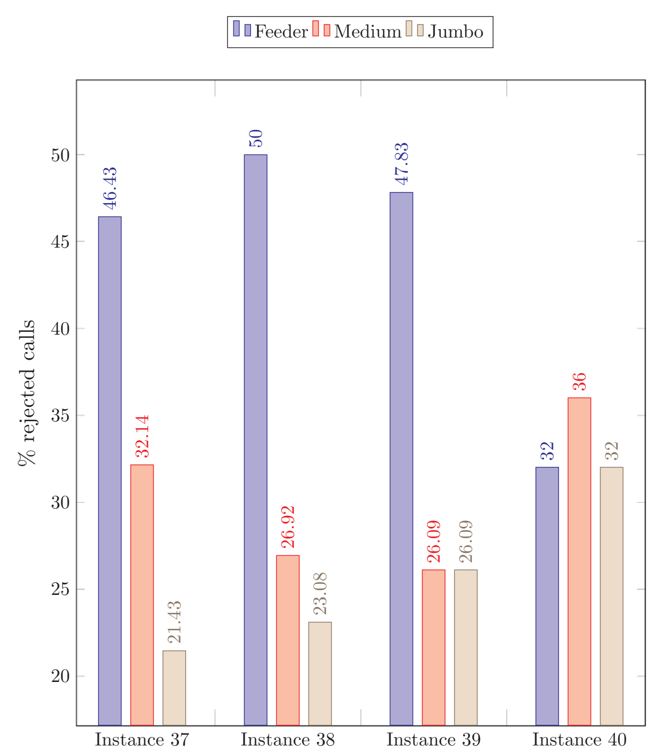

Fig. 2. Rejected calls for scenarios with 33% feeder, 33% medium, and 33% jumbo ships.

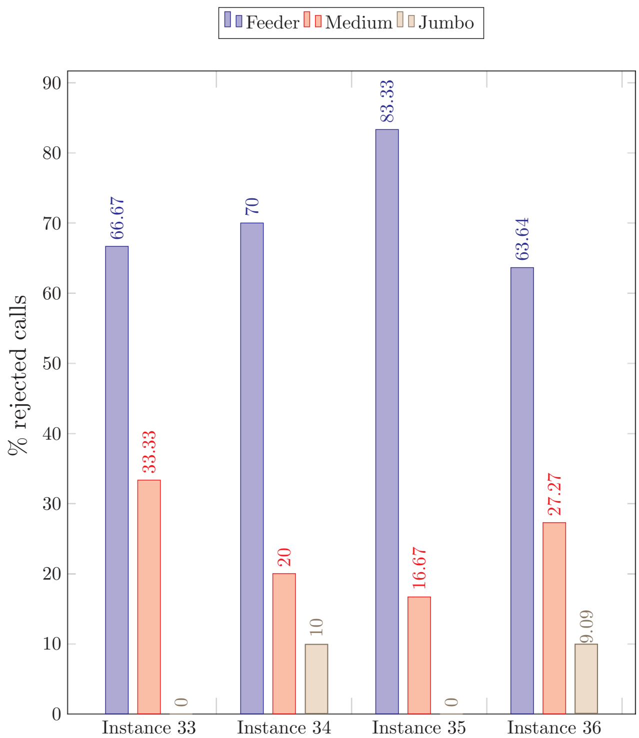

Fig. 3. Rejected calls for scenarios with 60% feeder, 30% medium, and 10% jumbo ships.

to the average upper bounds ($\overline{UB}$), accepted ship calls ($\overline{Acc.}$) and rejected ship calls ($\overline{Rej.}$). As can be observed in the table, in the smallest set of instances, the optimization models provide similar results in terms of accepted and rejected calls. However, the objective function values differ. This indicates that by using GSP and GSP+, we are able to provide better schedules for the same scenario, in the same solver under the same computational conditions. On the other hand, in the other cases SBTP+, GSP, and GSP+ report a similar number of accepted and rejected ships. However, it can be noticed that in those cases GSP+ accepts slightly fewer ships. This indicates that in those cases it may be beneficial to reduce the number of serviced ships while improving the quality of service. Finally, it can be seen that the overall computing requirement ($\overline{t(s.)}$) of GSP+ is much lower than the other models, and it exhibits such a performance in less than 2500 s.

Figs. 2 and 3 show the percentage of rejected calls for the scenarios considered: *(i)* a scenario where we have 33% feeder, 33% medium, and 33% jumbo ships, and *(ii)* 60% feeder, 30% medium, and 10% jumbo ships. When scenarios with limited resources (*i.e.* berths) are tackled, the feeder type ships are rejected in a higher percentage than in other cases, and the lowest amount of rejection corresponds to jumbo container ships. The rationale behind this is importance of jumbo ships in terms of connectivity with other types of ships due to their respective workload granting a major number of transshipments of container flows. It is important to note that in those cases where the jumbo ships arrive in a lower number to the terminal as reported in Fig. 3 the percentage of rejected calls is 0 or below 10%. Furthermore, it should be noted that there are some cases where the largest number of rejected calls is associated with medium ships. This reflects the complexity of this planning problem that not only depends on the percentage of the type of arriving ships but also on other characteristics such as the relationship among them, their service times, arrival times, etc.

5.2. *Problem instances considering 8 and 12 berths*

In this work, we also investigate the influence of having more berths and ships compared to those initially proposed in Imai et al. (2014). Tables 6–8 report the performance of SBTP+, GSP, and GSP+. In the tables, we report the upper bound (UB), the lower bound (LB), the relative error (Gap%) provided by CPLEX, and the computational time measured in seconds (t(s.)). The time to generate the assignments (columns) is less than 15 s for each instance, and this time is included in t(s.).

Based on the results reported in the tables, the following insights can be highlighted:

- Our GSP formulation outperforms the improved conceptual formulation (SBTP+) by providing the optimal solution for the majority of the instances. In many cases, it can be highlighted that both GSP and GSP+ formulations require a similar computational time. Moreover, the computational times are very competitive whereas the SBTP+ formulations always reach the time limit of 3 h.
- Although GSP already exhibits a very competitive performance, GSP+ is able to improve the quality of the lower bounds for instances (id 73–76) where GSP and GSP+ do not provide the optimal solution. It should be noted that for those 4 instances (id 73–76), GSP+ provides better solutions compared to GSP for three instances (id

Table 6
Computational results for the problem instances considering 70 ships and 8 berths.

			SBTP+				GSP				GSP+			
N	S	id	UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)	UB	LB	Gap(%)	t(s.)
70	8	49	441	0	100	10,800	39	39	0.00	28.97	39	39	0.00	26.40
		50	452	0	100	10,800	10	10	0.00	19.70	10	10	0.00	20.10
		51	365	0	100	10,800	8	8	0.00	18.84	8	8	0.00	19.90
		52	543	0	100	10,800	26	26	0.00	27.10	26	26	0.00	27.40
		53	881	0	100	10,800	32	32	0.00	26.30	32	32	0.00	28.10
		54	390	0	100	10,800	11	11	0.00	20.10	11	11	0.00	21.10
		55	611	0	100	10,800	22	22	0.00	27.70	22	22	0.00	28.60
		56	405	0	100	10,800	44	44	0.00	21.00	44	44	0.00	22.00
		57	499	0	100	10,800	0	0	0.00	15.00	0	0	0.00	16.00
		58	38	0	100	10,800	1	1	0.00	19.86	1	1	0.00	20.10
		59	559	0	100	10,800	0	0	0.00	16.70	0	0	0.00	18.10
		60	267	0	100	10,800	0	0	0.00	15.30	0	0	0.00	15.40
		61	66	0	100	10,800	0	0	0.00	15.30	0	0	0.00	16.20
		62	510	0	100	10,800	0	0	0.00	14.80	0	0	0.00	15.20
		63	67	0	100	10,800	0	0	0.00	14.20	0	0	0.00	14.30
		64	421	0	100	10,800	1	1	0.00	15.20	1	1	0.00	16.00
Average:			407.19	0	100	10,800	12.13	12.13	0.00	19.75	12.13	12.13	0.00	20.31

Table 7
Computational results for the problem instances considering 100 ships and 8 berths.

N	S	id	SBTP+ UB	SBTP+ LB	SBTP+ Gap(%)	SBTP+ t(s.)	GSP UB	GSP LB	GSP Gap(%)	GSP t(s.)	GSP+ UB	GSP+ LB	GSP+ Gap(%)	GSP+ t(s.)
100	8	65	3609	0	100.00	10,800	0	0	0.00	21.70	0	0	0.00	23.00
		66	3660	0	100.00	10,800	22	22	0.00	24.30	22	22	0.00	25.80
		67	4248	0	100.00	10,800	4	4	0.00	22.70	4	4	0.00	23.60
		68	2633	0	100.00	10,800	1	1	0.00	22.70	1	1.0	0.00	23.40
		69	2,568,475	0	100.00	10,800	74	74	0.00	28.20	74	74	0.00	29.20
		70	10,526	0	100.00	10,800	86	86	0.00	51.80	86	86	0.00	54.50
		71	1,421,464	0	100.00	10,800	304	304	0.00	1160.70	304	304	0.00	1172.30
		72	12,068	0	100.00	10,800	265	265	0.00	88.90	265	265	0.00	92.80
		73	736,699	160,000	78.28	10,800	161,850	386	99.76	10,800	160,662	160,317	0.21	10,800
		74	612,111	490,000	19.95	10,800	490,939	464	99.91	10,800	490,465	490,239	0.05	10,800
		75	376,421	80,000	78.75	10,800	80,892	688	99.15	10,800	80,743	80,631	0.14	10,800
		76	207,321	80,000	61.41	10,800	81,413	344	99.58	10,800	91,168	80,303	11.92	10,800
		77	1925	0	100.00	10,800	5	5	0.00	24.45	5	5.00	0.00	25.60
		78	2222	0	100.00	10,800	7	7	0.00	25.90	7	7	0.00	27.80
		79	2256	0	100.00	10,800	16	16	0.00	25.00	16	16	0.00	27.60
		80	3523	0	100.00	10,800	1	1	0.00	23.90	1	1	0.00	24.40
Average:			373072.56	50625.00	89.90	10,800	50992.44	166.66	24.90	2795	51488.94	50767.21	0.77	2796.88

Table 8
Computational results for the problem instances considering 150 ships and 12 berths.

N	S	id	SBTP+ UB	SBTP+ LB	SBTP+ Gap(%)	SBTP+ t(s.)	GSP UB	GSP LB	GSP Gap(%)	GSP t(s.)	GSP+ UB	GSP+ LB	GSP+ Gap(%)	GSP+ t(s.)
150	12	81	10,060,120	0	100	10,800	0	0	0.00	93.60	0	0	0.00	98.70
		82	4,975,601	0	100	10,800	6	6	0.00	68.90	6	6	0.00	86.20
		83	9,465,576	0	100	10,800	15	15	0.00	79.90	15	15	0.00	89.90
		84	5,780,425	0	100	10,800	20	20	0.00	86.10	20	20	0.00	100.30
		85	13,361,320	0	100	10,800	464	384	0.17	10,800	464	384	0.17	10,800
		86	17,400,000	0	100	10,800	362	362	0.00	3506.80	362	362	0.00	3511
		87	14,592,630	0	100	10,800	285	285	0.00	4633.10	285	285	0.00	4640.20
		88	17,420,010	0	100	10,800	298	165	0.45	10,800	297	165	0.45	10,800
		89	14,920,000	0	100	10,800	74	74	0.00	157	74	74	0.00	150
		90	14,890,000	0	100	10,800	87	87	0.00	274	87	87	0.00	273
		91	14,880,000	0	100	10,800	22	22	0.00	92	22	22	0.00	87
		92	15,020,000	0	100	10,800	312	312	0.00	221	312	312	0.00	220
		93	4,941,473	0	100	10,800	0	0	0.00	66	0	0	0.00	66
		94	8,630,000	0	100	10,800	0	0	0.00	62	0	0	0.00	62.20
		95	5,579,214	0	100	10,800	0	0	0.00	65.20	0	0	0.00	64.90
		96	4,027,991	0	100	10,800	1	1	0.00	64.60	1	1	0.00	64.90
Average:			10996522.50	0	100	10,800	121.63	108.30	0.04	1942.89	121.56	108.30	0.04	1944.68

Table 9
Mean values of average upper bounds ($\overline{UB}$), accepted ship calls ($\overline{Acc.}$), computational times ($\overline{t(s.)}$), and rejected ship calls ($\overline{Rej.}$) for instances considering 8 and 12 berths.

Set N	Set S	SBTP+ $\overline{UB}$	SBTP+ $\overline{t(s.)}$	SBTP+ $\overline{Acc.}$	SBTP+ $\overline{Rej.}$	GSP $\overline{UB}$	GSP $\overline{t(s.)}$	GSP $\overline{Acc.}$	GSP $\overline{Rej.}$	GSP+ $\overline{UB}$	GSP+ $\overline{t(s.)}$	GSP+ $\overline{Acc.}$	GSP+ $\overline{Rej.}$
70	8	407.19	10,800	70.00	0.00	12.13	19.75	70.00	0.00	12.13	20.31	70.00	0.00
100	8	373072.56	10,800	96.44	3.56	50992.44	2795.02	99.56	0.44	51488.94	2796.88	99.44	0.56
150	12	10996522.50	10,800	26.25	123.75	121.63	1941.89	150.00	0.00	121.56	1944.68	150.00	0.00
Average		3790000.75	10,800	64.23	42.44	17042.06	1585.55	106.52	0.15	17207.54	1587.29	106.48	0.19

73–75). Finally, on the large size instance set of 150 vessels and 12 berths in Table 8, it can be observed that GSP and GSP+ formulations obtain optimal solutions for many instances. Besides one problem instance (id 88), both GSP formulations exhibit a similar performance in terms of solution quality and time.

- In the case of larger scale instances, SBTP mostly reaches out-of-memory status before the models are loaded so that results are not reported in tables.

Similar to the previous subsection, in Table 9, each set of problem instances is compared according to the average upper bounds ($\overline{UB}$), accepted ship calls ($\overline{Acc.}$) and rejected ship calls ($\overline{Rej.}$). Even though SBTP+ does not provide the optimal solutions, the acceptance of calls is optimal as reflected by GSP and GSP+. On those instances considering 100 ships, GSP exhibits a better performance than GSP+ in terms of average upper bounds, computational time, and number of served ships. Nevertheless, this is because of the influence of the value

Table 10
Comparison of models in terms of their structure and generated nodes.

Set		SBTP				SBTP+				GSP				GSP+			
N	S	# bin.	# nonzero	# constr.	# nodes	# bin.	# nonzero	# constr.	# nodes	# bin.	# nonzero	# constr.	# nodes	# bin.	# nonzero	# constr.	# nodes
50	4	12,500	24710030.50	21,733	16879.88	12,500	584075.25	913	1328729.38	44046.25	2801396.25	1266.88	413153.125	44046.25	2810230.25	1265.875	519,899
70	4	42,323	95,687,518	24,500	3093.13	24,784	1538317.75	1277	87181.75	61943.25	3917300.50	1291.75	176855.688	61943.25	3942340.75	1290.75	373144.688
100	4	–	–	–	–	50,000	4344748.75	1824	4755.1875	88409.88	5,593,514	1324.94	18812.4375	88409.88	5665071.5	1323.9375	566493.313
70	8	–	–	–	–	44,100	3,063,969	2394	68969.5	123087.50	7,781,581	2505.75	0	123087.50	7,781,581	2505.75	0
100	8	–	–	–	–	90,000	8656093.50	3417	3711.5	177443.50	11,057,418	2515.50	37,381	177443.50	11,096,479	2515.75	33910.5
150	12	–	–	–	–	292,500	42708819.00	7527	16	400886.25	24,807,876	3766.75	4487.75	400886.25	24,807,876	3766.75	4480.625

in one instance where GSP is better than GSP+. In total, although both GSP formulations require around 1586 s on average, GSP+ provides more best solutions compared to GSP. Regarding accepted and rejected calls, it should be noted that both approaches provide a similar number. Meanwhile, the number of accepted ship calls is higher compared to scenarios with 4 berths. This is based on the fact that we can provide more service due to the extended berthing capacity.

5.3. Formulations comparison

This section is devoted to compare and analyze the performance of the models in terms of their structures and generated nodes. We present the average number of binary variables (# bin.), non-zeros (# nonzero), constraints (# constr.), and generated nodes (# nodes.) in Table 10.

As reported in the table, the number of constraints, binaries, and non-zeros is clearly reduced in the enhanced formulation SBTP+ with respect to SBTP. This reduction is translated into a better overall performance, allowing SBTP+ to generate more nodes compared to SBTP for the same time limit. Concerning the GSP formulations, it can be observed that adding lower bounds leads to generate more nodes in those cases where GSP+ performs better. It should be noted that in the case of 70 ships and 8 berths, both formulations provide the optimal solution without requiring any branching.

6. Conclusions

In this work, the strategic berth template problem (SBTP) has been investigated. Solving this problem supports terminals for better utilizing their berthing capacity within long planning horizons. It also gives a valuable input in the negotiations for the selection of ship calls where the optimization model allows to reject some calls. The problem has been addressed from a mathematical modeling perspective by providing and evaluating two mathematical formulations, one based on a three-index one and the other based on a GSP formulation.

For assessing the contributions of this work, a well-defined set of instances is proposed. Through it, we have evaluated the existing SBTP formulation (conceptually proposed in Imai et al. (2014)) in order to compare our extended formulation based on it. In this regard, we have found that this way of formulating the SBTP may not be appropriate for large instances. The contributions in terms of enhancements and additional lower bounds are significant as they allow to improve the quality of all solutions. They also enable the models to provide solutions for larger instances. Even more so, the GSP with and without additional bounds clearly outperforms the results of the unexplored formulation of Imai et al. (2014) in terms of computational time and solution quality. This justifies the use of these optimization models, depending on the scenario, instead of using heuristic approaches. Finally, having such strong formulations helps to conduct a posteriori analysis even if that involves additional computational time.

Based on the contributions of this work, the next stage of our investigation aims at extending this problem on different quay layouts such as continuous or hybrid. Another valuable extensions are the consideration of quay cranes assignment (Iris, Pacino, & Ropke, 2017; Lalla-Ruiz et al., 2014), yard capacity (Zhen et al., 2011) or waterway restrictions (Du, Chen, Lam, Xu, & Cao, 2015; Lalla-Ruiz, Shi, & Voß, 2018) with respect to the SBTP.

Acknowledgments

The authors would like to thank the area editor and two anonymous reviewers for insightful comments and suggestions. This study is supported by project Ref. M4061473 at Nanyang Technological University and partially funded by the Spanish Ministry of Economy and Competitiveness with FEDER funds (project TIN2015- 70226-R).

References

Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research, 202*, 615–627.

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*, 675–689.

Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J., & Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison. *Transportation Research Part E: Logistics and Transportation Review, 47*, 461–473.

Christensen, C. & Holst, C. (2008). Berth allocation in container terminals. Master's thesis. Department of Informatics and Mathematical Modelling, Technical University of Denmark [In Danish].

Cordeau, J. F., Laporte, G., Legato, P., & Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem. *Transportation Science, 39*, 526–538.

Du, Y., Chen, Q., Lam, J. S. L., Xu, Y., & Cao, J. X. (2015). Modeling the impacts of tides and the virtual arrival policy in berth allocation. *Transportation Science, 49*, 939–956.

Gerstl, E., & Mosheiov, G. (2012). Scheduling on parallel identical machines with job-rejection and position-dependent processing times. *Information Processing Letters, 112*, 743–747.

Giallombardo, G., Moccia, L., Salani, M., & Vacca, I. (2010). Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological, 44*, 232–245.

Hendriks, M., Armbruster, D., Laumanns, M., Lefeber, E., & Udding, J. T. (2012). Strategic allocation of cyclically calling vessels for multi-terminal container operators. *Flexible Services and Manufacturing Journal, 24*, 248–273.

Hoffarth, L., & Voß, S. (1994). Liegeplatzdisposition auf einem Container Terminal - Ansätze zur Entwicklung eines entscheidungsunterstützenden Systems. In H. Dyckhoff, U. Derigs, M. Salomon, & H. Tijms (Eds.). *Operations research proceedings 1993* (pp. 89–95). Berlin: Springer.

Hu, Z.-H. (2015). Multi-objective genetic algorithm for berth allocation problem considering daytime preference. *Computers & Industrial Engineering, 89*, 2–14.

Huang, K., Suprayogi, & Ariantini (2016). A continuous berth template design model with multiple wharfs. *Maritime Policy & Management, 43*, 763–775.

Imai, A., Nishimura, E., & Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological, 35*, 401–417.

Imai, A., Nishimura, E., & Papadimitriou, S. (2003). Berth allocation with service priority. *Transportation Research Part B: Methodological, 37*, 437–457.

Imai, A., Yamakawa, Y., & Huang, K. (2014). The strategic berth template problem. *Transportation Research Part E: Logistics and Transportation Review, 72*, 77–100.

Iris, C., Pacino, D., & Ropke, 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.

Iris, C., Pacino, D., Ropke, 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.

Jans, R. (2009). Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *INFORMS Journal on Computing, 21*, 123–136.

Jin, J. G., Lee, D.-H., & Hu, H. (2015). Tactical berth and yard template design at container transshipment terminals: A column generation based approach. *Transportation Research Part E: Logistics and Transportation Review, 73*, 168–184.

Kim, K. H., & Moon, K. C. (2003). Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological, 37*, 541–560.

Lalla-Ruiz, E., Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, J. M. (2016). A set-partitioning-based model for the berth allocation problem under time-dependent limitations. *European Journal of Operational Research, 250*, 1001–1012.

Lalla-Ruiz, E., González-Velarde, J. L., Melián-Batista, B., & Moreno-Vega, J. M. (2014). Biased random key genetic algorithm for the tactical berth allocation problem. *Applied Soft Computing, 22*, 60–76.

Lalla-Ruiz, E., Melián-Batista, B., & Moreno-Vega, J. M. (2012). Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem. *Engineering Applications of Artificial Intelligence, 25*, 1132–1141.

Lalla-Ruiz, E., Shi, X., & Voß, S. (2018). The waterway ship scheduling problem. *Transportation Research Part D: Transport and Environment, 60*, 191–209.

Lalla-Ruiz, E., & Voß, S. (2016). POPMUSIC as a matheuristic for the berth allocation problem. *Annals of Mathematics and Artificial Intelligence, 76*, 173–189.

Lalla-Ruiz, E., Voß, S., Expósito-Izquierdo, C., Melián-Batista, B., & Moreno-Vega, J. M. (2017). A POPMUSIC-based approach for the berth allocation problem under time-dependent limitations. *Annals of Operations Research, 253*, 871–897.

Lu, L., Zhang, L., & Yuan, J. (2008). The unbounded parallel batch machine scheduling with release dates and rejection to minimize makespan. *Theoretical Computer Science, 396*, 283–289.

Mokotoff, E. (2001). Parallel machine scheduling problems: A survey. *Asia-Pacific Journal of Operational Research, 18*, 193–242.

Monaco, M. F., & Sammarra, M. (2007). The berth allocation problem: A strong formulation solved by a Lagrangean approach. *Transportation Science, 41*, 265–280.

Moorthy, R., & Teo, C.-P. (2006). Berth management in container terminal: The template design problem. *OR Spectrum, 28*, 495–518.

Nishimura, E., Imai, A., & Papadimitriou, S. (2001). Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research, 131*, 282–292.

Ribeiro, G. M., Mauri, G. R., de Castro Beluco, S., Lorena, L. A. N., & Laporte, G. (2016). Berth allocation in an ore terminal with demurrage, despatch and maintenance. *Computers & Industrial Engineering, 96*, 8–15.

Shirvani, N., Ruiz, R., & Shadrokh, S. (2014). Cyclic scheduling of perishable products in parallel machine with release dates, due dates and deadlines. *International Journal of Production Economics, 156*, 1–12.

United Nations Conference on Trade and Development. (2014). *Review of maritime transport*. <http://www.unctad.org>.

Venturini, G., Iris, C., Kontovas, C. A., & 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.

Wang, S., Liu, Z., & Qu, X. (2017). Weekly container delivery patterns in liner shipping planning models. *Maritime Policy & Management, 44*, 442–457.

Wang, S., Meng, Q., & Lee, C.-Y. (2016). Liner container assignment model with transit-time-sensitive container shipment demand and its applications. *Transportation Research Part B: Methodological, 90*, 135–155.

Xu, D., Li, C. L., & Leung, J. Y. T. (2012). Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research, 216*, 47–56.

Yalaoui, F., & Chu, C. (2006). New exact method to solve the $Pm/rj/\sum Cj$ schedule problem. *International Journal of Production Economics, 100*, 168–179.

Zhang, L., & Lu, L. (2016). Parallel-machine scheduling with release dates and rejection. *4OR, 14*, 165–172.

Zhen, L., Chew, E. P., & Lee, L. H. (2011). An integrated model for berth template and yard template planning in transshipment hubs. *Transportation Science, 45*, 483–504.