Contents lists available at ScienceDirect

### European Journal of Operational Research

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

### **Discrete Optimization**

# New formulations and solutions for the strategic berth template problem

### Elena Fernández, Manuel Munoz-Marquez\*

Departamento de Estadística e Investigación Operativa, Universidad de Cádiz, Spain

### ARTICLE INFO

Article history: Received 10 December 2020 Accepted 29 June 2021 Available online 7 July 2021

Keywords: Combinatorial optimization Maritime transportation Strategic berth allocation

### ABSTRACT

This paper develops new formulations for the Strategic Berth Template Problem, which combines strategic and operational decisions for medium-term berth planning of a given set of cyclically calling ships. The strategic decisions determine the ship calls that will be served, whereas the operational ones establish the berth template that will be applied in a cyclic fashion in the planning horizon. The proposed formulations use binary variables that classify served ships depending on whether or not their service starts in their arrival cycle or in the next one. This helps modeling the problem, since a closed linear expression can be obtained for the waiting times. Constraints imposing that the availability of the berths is respected at each time period can be derived by defining additional binary variables pointing to the starting service times of the served ships. Aggregating such variables over all berths leads to a relaxed formulation, which can be solved in remarkably small computing times. Furthermore, the solution of an auxiliary subproblem produces feasible solutions to the original problem as well as a simple optimality check. Disaggregating the initial service time variables for the different berths leads to a valid formulation. Numerical results from extensive computational tests over a set of benchmark instances from the literature are presented and analyzed. The obtained results assess the excellent performance of the proposed formulations, which outperform existing ones.

© 2021 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

### 1. Introduction

In this paper we study the Strategic Berth Template Problem (SBTP). The SBTP combines strategic and operational decisions for medium-term berth planning of a given set of cyclically calling ships. While the strategic decisions dictate the ship calls that will be served and those whose service call will be rejected, the operational decisions determine the berth template that will be applied in a cyclic fashion in the considered planning horizon. Furthermore, there may be links relating the strategic decisions of service to ships belonging to certain groups. These links are derived from strong transhipment relations between some large-size mother ships and some smaller feeder ships, which are contractually attached to each other. All the ships within each group must be handled similarly, in the sense that all of them are either served or rejected.

In particular, the SBTP aims at deciding which calling ships should be accepted for berthing, and determines the most ap-

\* Corresponding author.

propriate berth/time allocation for the accepted incoming traffic. Specifically, its aim is to develop a template for the accepted ships for a cyclic time horizon, consisting of a berth allocation together with a service (berthing) time-window for each of the accepted ships. The following issues must be taken into account: (*i*) there is a limited number of available berths; (*ii*) due to the cyclic nature of the template, the service time-windows of the ships allocated to the same berth must be non-overlapping; and (*iii*) service to a ship may start in the next cycle to the one when it arrives to the port, or, even if service starts in the same cycle when the ship arrives, its service may terminate in the next cycle. The objective is to minimize the sum of the waiting times of the accepted ships plus a penalty for each rejected call proportional to its workload.

Broadly speaking, berth allocation problems (BAP) aim at assigning berthing positions and service times to calling ships at a container terminal. Different variants of such problems have been studied in the literature. The reader is addressed to Bierwirth & Meisel (2010, 2015) for surveys on the topic, or to the inspiring introduction and literature review of Iris, Lalla-Ruiz, Lam, & Voss (2018) where the relevance and actual economic implications of these problems are highlighted, and the different perspectives and ingredients that BAPs may integrate are motivated and overviewed.





*E-mail addresses:* elena.fernandez@uca.es (E. Fernández), manuel.munoz@uca.es (M. Munoz-Marquez).

<sup>0377-2217/© 2021</sup> The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Examples of BAP that address tactical or operational decisions related to ships assignment and service are (Buhrkal, Zuglian, Ropke, Larsen, & Lusby, 2011; Cordeau, Laporte, Legato, & Moccia, 2005; Imai, Nishimura, & Papadimitriou, 2003; Lalla-Ruiz & Voss, 2016; Monaco & Sammarra, 2007; Xu, Li, & Leung, 2012), to mention just a few. Balancing the workload distribution over time was considered in Jin, Lee, & Hu (2015), who addressed the quayside berthing congestion from a tactical planning viewpoint. For this, the authors jointly addressed the berth template design problem and the yard template design problem with an objective combining costs due to container flows with those dealing with quay-side workload imbalance.

As indicated in Iris et al. (2018), liner shipping companies usually call each port on the same day of every week (see also Wang & Lee, 2016; Wang & Qu, 2017). Hence, service contracts between shipping lines and ports require a template for berthing, i.e. previously allocated berthing slots, which can be cyclically (weekly) repeated throughout the term of the contract so container shipping services to each ship are provided according to the template on a fixed day of each week. These circumstances motivated the study of the Berth Template Problem (BTP), which focuses on finding a template for a berth plan of fixed length (typically one week), which is used cyclically over a long-term horizon. The essential difference between the BTP and the BAP is that the fixed planning horizon is repeated in a cyclic fashion. Hence BTP models take into account that service to a vessel scheduled close to the end of the planning horizon is likely to extend into the next planning period. That is, the BTP is considered as a packing problem on a cylinder. Such a cyclical allocation of calling ships at multiple terminals within the same port was first studied in Hendriks, Laumanns, Lefeber, & Udding (2010); Imai, Yamakawa, & Huang (2014); Moorthy & Teo (2006); Zhen & Chang (2012). The BTP has also been studied for a continuous quay as a mid-term tactical decision problem in Huang, Suprayogi, & Ariantini (2016).

On the other hand, the limited weekly berthing capacity of ports makes it necessary to face with models integrating strategic decisions on the calls that must be accepted/rejected. Still most BTPs studied in the literature ignore such strategic decisions. Imai et al. (2014) introduced the SBTP, which is defined on a strategic planning level and integrates decisions regarding the selection of ship calls to be served with the assignment of berth time-windows for selected ships within a cyclic horizon. One of the modeling assumptions of the SBTP is that the length of the cycle is the same for all calls. Furthermore, the SBTP incorporates additional conditions linking the acceptance/rejection decisions of mother and feeder ships under consideration. Imai et al. (2014) proposed a formulation extending the formulation of the dynamic berth allocation problem of (Imai, Nishimura, & Papadimitriou, 2001) and several heuristics based on the solution of a Lagrangean dual with alternative subgradient optimization approaches. The computational results showed the difficulty for solving the problem with the proposed heuristics. The SBTP has also been studied by Iris et al. (2018), who analyzed the initial formulation proposed in Imai et al. (2014), which remained computationally unexplored, and proposed a different formulation based on the solution of a generalized setpacking problem (GSP). Both formulations were reinforced by including additional lower bounds. A set of benchmark instances was created and used in the extensive computational experiments carried out. The obtained results showed that both formulations notably improved with the addition of the lower bounds and highlighted the superiority of the reinforced GSP formulation.

In this paper we focus on the SBTP, and develop new mixedinteger linear programming formulations and algorithmic alternatives for solving it. In addition to the *natural* strategic binary variables associated with the acceptance/rejection of ship calls, all the proposed formulations use binary variables that classify served ships depending on whether or not their service starts during their arrival cycle or in the next one. This helps modeling the STBP, since a closed linear expression can be obtained for the waiting times. The most basic formulations use additional variables that relate served ships with their immediate predecessors in the corresponding berths. Still, such variables can be avoided by expressing the starting times of service to ships in terms of new binary decision variables indicating whether or not their service starts at the different time periods of the planning horizon. In its turn, these new binary variables allow us to count the number of ships being served simultaneously at each time period, providing us with the possibility of guaranteeing that the availability of berths is respected at each time period. Aggregating the new decision variables over all berths leads to a formulation for a relaxation of SBTP, which can be solved in remarkably small computing times. Moreover, the solution to an auxiliary subproblem reveals if the relaxed aggregated solution can be disaggregated to a feasible SBTP solution. This leads to a simple feasibility check indicating whether or not the solution at hand is optimal for the SBTP. Therefore, the aggregated formulation can be combined with the feasibility check within a 2-phase solution algorithm for the SBTP. Alternatively, considering disaggregated variables for the initial time periods for service to accepted ships at the different berths produces a valid formulation for the SBTP, at the expenses of increasing its total number of binary variables and constraints. Still the formulation can be solved very efficiently with any off-the-shelf solver and produces excellent results.

Extensive computational tests have been carried out with the set of 96 benchmark instances generated in Iris et al. (2018) with a number of calling ships in {50, 70, 100, 150}, and a number of berths in {4, 8, 12}. The obtained results highlight the effectiveness of the two formulations based on the indicator variables for the time periods when service to accepted ships start. Both the 2-phase solution algorithm based on the relaxed formulation with the aggregated variables, as well as the exact formulation using the disaggregated decision variables outperform the most efficient formulation proposed in Iris et al. (2018). The 2-phase algorithm has solved to proven optimality for 78 out of the 96 considered benchmark instances in computing times that are always below 500 seconds. The disaggregated formulation was able to solve 94 benchmark instances within a maximum time limit of three hours, and produced very small percentage optimality gaps for the remaining two instances.

This paper contributes to the study of the SBTP introducing a new class of formulations using binary variables that allow to get a closed linear expression for waiting times. These formulation are solved very efficiently with any off-the-shelf solver. Also a relaxation of the problem is introduced that produce very tight lower bounds in very small computing times.

From algorithmically point of view, a two phase algorithm is introduced that solves a relaxed version of the problem and then makes simply feasibility test, giving the solution or proceeding to solve the non-relaxed formulation.

The remainder of this paper is structured as follows. In Section 2 we give a formal definition of the SBTP and discuss its relation to some well-known combinatorial problems. Section 3 introduces the basic formulations where waiting times are expressed in terms of the decision variables indicating whether service to an accepted call starts in its arrival cycle or in the next one. In Section 3.1 we introduce a formulation with the aggregated service-start time period variables, whereas in Section 3.2 we show that it is a relaxation of the SBTP and study some of its properties that will be exploited in the design of the 2-phase algorithm. Sections 4 and 4.1 introduce the valid SBTP formulation based on the disaggregated service-start time period variables, and give a comparison of all the developed formulations in terms of

their number of decision variables and constraints. In Section 5 we deal with several algorithmic issues that will be exploited for determining feasible solutions, either from scratch or from the relaxed aggregated formulation. The computational tests that have been carried out are described in Section 6 where we summarize and analyze the numerical results that we have obtained, and compare them with those of Iris et al. (2018). The paper closes in Section 7 with some conclusions and final remarks.

### 2. The strategic berth template problem

As mentioned, the aim of the SBTP is to develop a template for a cyclic time horizon, in which service calls of some ships may be rejected. Such a template must differentiate among accepted (served) and rejected (non-served) ships. Furthermore, it must take into account that service to an accepted ship may start in the next cycle to the one when it arrives to the port, or that, even if service starts in the same cycle when the ship arrives, its service may terminate in the next cycle. In order to differentiate among the potentially different cycles when operations take place, we will use the terms arrival cycle, service cycle, and termination cycle to refer to the cycle when a given ship arrives in port, its service starts, and its service terminates, respectively. When the service cycle of a served ship does not coincide with its arrival cycle, it will be the cycle next to the arrival cycle of the ship. Similarly, when the termination cycle of a served ship does not coincide with its service cycle, it will be the cycle next to the service (and arrival) cycle of the ship.

Let *H* denote the duration of a cycle and  $V = \{1, ..., n\}$  the index set for ships. Associated with each ship  $i \in V$ , let  $a_i$  and  $c_i$  denote its arrival and processing (service) time, respectively. We assume that within each cycle all activities take place in a discretized time horizon  $T = \{1, ..., H\}$ . Thus, we assume that  $a_i, c_i$ , as well as the times of all operations related to accepted ships take values in *T*. Let  $K = \{1, ..., m\}$  denote the index set for the linked subgroups of ships,  $V_k \subset V$  the *k*-th subset of ships, and  $C_k = \sum_{i \in V_k} c_i$  the total processing time of all the ships in mother-ship class *k*. By defining a singleton subset  $V_k = \{i\}$  (with associated coefficient  $C_k = c_i$ ) for every ship *i* originally not attached to any mother-ship, we assume without loss of generality that  $\{V_k\}_{k \in K}$  defines a partition of *V*. In the following, for each  $i \in V$ ,  $k(i) \in K$  denotes the index of the subset  $V_k$  such that  $i \in V_{k(i)}$ .

The objective of the SBTP is twofold. On the strategic side, it aims at reducing service call rejections. On the operational side, it aims at reducing the times that (served) ships wait since their arrival until their service starts. Specifically, let  $g \ge 0$  be a given penalty per rejected unit service time; that is, if the call of mothership class  $k \in K$  is rejected, then a cost  $g \times C_k$  is incurred. In addition, each served ship incurs a cost of one unit per unit of waiting time. Therefore, the objective consists of the sum of two terms: (*i*) the total penalty for rejected calls  $(g \sum_{k \in \overline{K}} C_k)$ , and (*ii*) the total waiting time of served ships  $(\sum_{i \in \overline{V}} w_i)$ , where  $\overline{K} \subseteq K$  denotes the index set of rejected mother-ship classes,  $\overline{V} \subseteq V$  the index set of served ships, and  $w_i$  the time that ship *i* waits since its arrival until its service starts.

The SBTP is to determine a partition of mother-ship classes to be served/rejected, as well as an allocation to berths together with a cyclic sequence for service to the accepted calls allocated to the same berth, such that the overall service time of the calls served in the same berth does not exceed the duration of the cycle, H, of minimum total cost.

Observe that the SBTP integrates three difficult combinatorial problems. On the one hand, the selection of the ships that will be served, respecting the relations among the ships in each group. On the other hand, the allocation of accepted ships to berths, respecting the cycle duration within each berth, which can be seen as a bin packing problem (Korte & Vygen, 2006). Finally, the optimal sequencing of service to all the ships allocated to each berth, which can be reduced to the problem of finding the service schedule that minimizes the total tardiness (Du & Leung, 1990). Both the bin packing and minimizing tardiness are already NP-hard problems.

### 3. Mathematical programming formulations for the SBTP

In this section we develop several mathematical programming formulations for the SBTP. All of them use binary decision variables to determine the strategic decisions on ship calls that are served/rejected:

•  $z_k \in \{0, 1\}, k \in K. \ z_k = 1 \iff$  the ships in mother-ship class k are served.

Since the objective function depends on the waiting times of served ships, which, in turn, depend on their service starting times, we define the following additional decision variables:

- $s_i$ : service starting time (or just starting time) of ship  $i \in V$ .
- $w_i$ : waiting time of served ship  $i \in V$ . This is the time since the ship arrived at the terminal and its service started.

Using variables z and w the objective function can be written as

min 
$$g\sum_{k\in K}C_k(1-z_k) + \sum_{i\in V}w_i.$$
 (1a)

The conditions that regulate the relationship among the above variables, and their relationship with the cycle length depend on whether or not the arrival, service and termination cycles of the involved served ships coincide. For instance, for ships whose arrival and service cycles coincide, it holds that  $s_i \ge a_i$ . On the contrary, this lower bound on the value of  $s_i$  is no longer valid for ships whose service starts in the next cycle after their arrival, for which  $s_i \le a_i - 1$  must hold to guarantee that the duration of the cycle is respected. For such ships, taking into account that the duration of the cycle must include service times, the above bound can be reinforced to  $s_i \leq a_i - c_i$ . A similar observation can be made with respect to the waiting times. While the duration of the cycle imposes that  $w_i \leq H - a_i - c_i$  for the ships whose arrival and service cycles coincide, for ships with different arrival and service cycles we have that  $w_i \ge H - a_i$ . The starting and waiting times of non-served ships will be zero.

The above observation indicates that, in order to compute accurately the waiting times derived from feasible service schedules, additional information is needed indicating whether or not the service cycle of each served ship coincides with its arrival cycle. To this end, associated with each  $i \in V$  we define two new complementary binary variables  $x_i$  and  $y_i$ , where  $x_i = 1$  (and  $y_i = 0$ ) if and only if ship *i* is served and its service cycle coincides with its arrival cycle, whereas  $x_i = 0$  (and  $y_i = 1$ ) if and only if ship *i* is served but its service cycle is the cycle next to its arrival cycle. Therefore,

$$z_{k(i)} = x_i + y_i \qquad \qquad i \in V \qquad (2a)$$

$$w_i = s_i - a_i z_{k(i)} + H y_i \qquad i \in V.$$
(2b)

Constraints (2a) also guarantee that the mother-ship relationship of the ships in each class is respected, and that for any non-served ship,  $x_i = y_i = 0$ . Therefore, for non-served ships, Constraints (2b) reduce to  $w_i = s_i$ , i.e. the waiting times of rejected ships coincide with their starting times. Taking into account the minimization objective function and that there are no lower



**Fig. 1.** Example of definition of variables with n = 5, H = 10 and b = 2.

bounds for the starting times of non-served ships (see also constraints (3a)-(3c) below), in any optimal solution for any non-served ship it will hold that  $w_i = s_i = 0$ .

The transition between consecutive cycles is controlled through an additional set of binary variables  $f_i$ ,  $i \in V$  such that  $f_i = 1$  if and only if ship *i* is served and its service cycle does not coincide with its termination cycle. This means that the service cycle of ship *i* coincides with its arrival cycle ( $x_i = 1$ ) but it does not coincide with its termination cycle ( $s_i + c_i > H$ ). Moreover, with the aid of variables  $f_i$ ,  $i \in V$  we can also relate the starting times of served ships with variables *x* and *y*. That is:

$$f_i \le x_i$$
  $i \in V$  (3a)

$$(H+1)f_i \le s_i + c_i z_{k(i)} \qquad \qquad i \in V \tag{3b}$$

$$a_i x_i + y_i \le s_i \le (H - c_i) x_i + c_i f_i + (a_i - c_i) y_i \quad i \in V.$$
 (3c)

Fig. 1 illustrates the definition of the above variables with an example with two berths, five calling ships, singleton mother-ship classes, and a time horizon H = 10. Arrival and service times are given, by a = (1, 1, 5, 9, 10) and c = (3, 3, 3, 3, 2), respectively. The figure shows a solution in which ships  $\{1, 4, 5\}$  are served in berth 1 (in that order) and ships  $\{2, 3\}$  in berth 2. Since all the ships are served, we have  $z_k = 1$ ,  $k \in \{1, \dots, 5\}$ . In the displayed solution, the starting times are given by s = (4, 1, 5, 9, 2), with associated waiting times w = (3, 0, 0, 0, 2). Since  $a_i \le s_i$ , for,  $i \in \{1, 2, 3, 4\}$  we have  $x_1 = x_2 = x_3 = x_4 = 1$ , as their service and arrival cycles coincide; instead,  $s_5 = 2 < a_5 = 10$ , which means that service to ship 5 starts in the next cycle to the one when it arrives so  $y_5 = 1$ . Furthermore,  $f_4 = 1$  since its service cycle does not coincide with its termination cycle.

In order to obtain the actual schedule of each berth and to guarantee that the overall processing time of all the ships served in the same berth does not exceed the cyclic time horizon H, we define additional *predecessor* variables, which, even if they do not give an explicit allocation of ships to berths, permit determining the service sequence at the berths, since they define the order in which ships are served in each cycle and, implicitly, they define clusters of ships served cyclically in the same berth. In particular let:

•  $p_{ij} \in \{0, 1\}$ ,  $i, j \in V$ . When  $i \neq j$ ,  $p_{ij} = 1 \iff$  ships *i* and *j* are processed consecutively in the same berth and ship *j* is served immediately before ship *i*, allowing that there is some idle time in the berth after termination of *j*. When  $p_{ij} = 1$  we will indistinctively say that *j* is the *predecessor* of *i*, *j precedes i*, or *i follows j*. When i = j,  $p_{ii} = 1$  means that ship *i* is its own *predecessor*, that is, *i* is designated as the *first* ship processed in its berth.

The constraints regulating the service sequence of each berth are:

$$\sum_{i \in V} p_{ij} = z_{k(i)} \qquad i \in V$$
(4a)

$$\sum_{j \in V \setminus \{i\}} p_{ji} \le z_{k(i)} \qquad i \in V$$
(4b)

$$\sum_{i\in V} p_{ii} \le b. \tag{4c}$$

Constraints (4a) impose that each served ship has a unique predecessor, whereas (4b) indicate that each served ship can precede at most one ship. The last ship served in each berth in each cycle will not precede any other ship. Constraints (4a) also guarantee that all the ships in each mother-ship class are either served or rejected. By constraint (4c) no more than *b* berths are used.

Note that due to the cyclic nature of berth schedules, with the above predecessor variables there can be multiple representations of the berth schedule associated with a given sequence of ships. The only difference among all the equivalent representations is, in essence, the ship of the sequence that is designated as the *first* ship processed in the berth. Any served ship can be selected as the first ship in its berth and the time period when its service starts can be used as the reference to ensure that the duration of the sequence of all the ships served in that berth does not exceed the cycle duration *H*. For instance, in the example of Fig. 1 we could chose  $p_{11} = p_{41} = p_{54} = 1$ ,  $p_{22} = p_{32} = 1$ .

We define a final set of decision variables associated with the times where some events take place:

- $e_i$ : completion time of ship  $i \in V$ , i.e. the time when service to ship  $i \in V$  has been completed. This is the first period of time when the berth is already available for serving the next ship in the sequence of the berth.
- $v_i$ : idle time of the berth where ship  $i \in V$  is served, immediately before its service starts. This is the time since the completion of service to the predecessor of i and the arrival at the terminal of ship i.
- *o<sub>i</sub>*: time that the berth that serves ship *i* has been occupied since the beginning of service to the first ship in the berth until service to ship *i* starts.

Together with the relations that determine the precise values of variables  $e_i$ , the pairs of ships processed consecutively in the same berth must satisfy the following sets of constraints:

$e_i = s_i + c_i z_{k(i)} - H f_i$	$i \in V$	(5a)
$s_i \ge e_j - H(1 - p_{ij})$	$i, j \in V, i \neq j$	(5b)
$w_i \ge e_j - a_i - H(1 - p_{ij})$	$i, j \in V, i \neq j$	(5c)
$v_i \ge s_i - e_j - H(1 - p_{ij})$	$i, j \in V, i \neq j$	(5d)

$$o_i \ge o_j + c_j + v_i - H(1 - p_{ij})$$
  $i, j \in V, i \neq j.$  (5e)

Constraints (5b)–(5e) only become active for pairs of ships  $i, j \in V, i \neq j$ , such that  $p_{ij} = 1$ . In particular, (5b) establish that, if *i* follows *j*, then service to ship *i* cannot start before termination of service to ship *j*. Similarly, (5d) impose that, if *i* follows *j*, the idle time immediately before service to ship *i* must be at least the difference between the starting time of *i* and the completion time of *j*. The occupation times of consecutive ships in the same berth are regulated by (5e).

The inequalities (6a) and (6b) below set upper bounds for occupation and idle times, respectively.

$$\nu_i \le a_i x_i \qquad \qquad i \in V \tag{6a}$$

$$o_i \le (H - c_i)(z_{k(i)} - p_{ii})$$
  $i \in V.$  (6b)

Note that occupation times reduce to zero for non-served ships as well as for the first ships served in each berth. The upper bounds  $H - c_i$ ,  $i \in V$ , guarantee that the cycle duration is respected. Non-zero idle times can only appear for ships whose arrival and service cycles coincide (otherwise the ship will be served as soon as the berth becomes available in the next cycle). That is, for each berth, we implicitly set the beginning of its cycle to the starting time of the first ship served in the berth.

It can be easily checked that in the running example of Fig. 1, the values of these variables are e = (7, 4, 8, 2, 4), v = (0, 0, 1, 2, 0), o = (0, 0, 4, 5, 8).

Therefore a valid formulation for the SBTP is:

F0 min 
$$\sum_{i \in V} w_i + g \sum_{k \in K} C_k (1 - z_k)$$
 (1a)

$$z_{k(i)} = x_i + y_i \qquad \qquad i \in V \qquad (2a)$$

$$w_i = s_i - u_i z_{k(i)} + H y_i \qquad l \in V \qquad (2D)$$
  
$$f \in Y \qquad i \in V \qquad (22)$$

$$\begin{array}{ll} f_i \geq x_i & i \in V \\ (H+1)f_i \leq s_i + c_i z_{k(i)} & i \in V \end{array} \tag{3b}$$

$$a_i x_i + y_i \le s_i \le (H - c_i) x_i + c_i f_i + (a_i - c_i) y_i \ i \in V$$
 (3c)

$$\sum_{i \in V} p_{ij} = z_{k(i)} \qquad \qquad i \in V \qquad (4a)$$

$$\sum_{i \in V \setminus \{i\}} p_{ji} \le z_{k(i)} \qquad \qquad i \in V \qquad (4b)$$

$$\sum_{i \in V} p_{ii} \le b \tag{4c}$$

Formulation F0 can be reinforced by adding tighter lower and upper bounds on starting times, waiting times and termination times (see (8a) and (8b) below), which reduce to zero for non-served ships:

$$(H - a_i + 1)y_i \le w_i \le (H - c_i)z_{k(i)} \qquad i \in V$$
(8a)

$$(a_i + c_i)x_i - Hf_i + (1 + c_i)y_i \le e_i \le H(x_i - f_i) + c_if_i + a_iy_i$$
  $i \in V.$  (8b)

Unfortunately, despite the above reinforcements (or other of similar nature), the Linear Programming (LP) bounds of formulation F0 tend to be very weak, which is due to the *Big-M* type of Constraints (5b)–(5e). For this reason, in the next sections we develop other formulations in which these constraints can be removed, at the expenses of introducing additional sets of binary decision variables.

### 3.1. Counting the number of ships served at a given time period

One of the main difficulties of the SBTP is to control in an effective way that the number of ships being served simultaneously at any time period  $t \in T = \{1, ..., H\}$  does not exceed the available number of berths. With the current set of decision variables the limitation on the maximum number of available berths is only controlled via constraint (4c), which counts the overall number of "first ships". In this section we introduce an additional set of decision variables that allows us to obtain a linear expression for the number of served ships at any time period. For a given solution s, let us identify the set of served ships whose status is "being served" at a given time period  $t \in T$ . This set may contain ships whose service cycle coincides with the current cycle (that is,  $s_i \leq t$ ) as well as ships whose service started in the cycle previous to that of t (that is,  $t < s_i$ ). We denote by  $Z^{t=}(s)$  and  $Z^{t-}(s)$  the index set of ships of each of these two clases whose service remains active at time period t, respectively. In particular,  $Z^{t=}(s)$  consists of the indices of all served ships with  $s_i \le t$ such that  $s_i + c_i - 1 \ge t$ , whereas  $Z^{t-}(s)$  consists of the indices of all served ships with  $s_i > t$  whose service remains active at time period *t* of the following cycle, i.e.  $s_i + c_i - 1 - H \ge t$ . While only ships with  $c_i > t$  may appear in set  $Z^{t^-}(s)$ , the set  $Z^{t^-}(s)$ may contain indices of ships with both  $c_i \le t$  and  $c_i > t$ . Therefore, taking into account that  $1 \le s_i \le H$ , the above two sets are given by  $Z^{t^{=}}(s) = \{i \in V \mid s_i \in [\max\{1, t - c_i + 1\}, t]\}$ , and  $Z^{t^{-}}(s) =$  $\{i \in V \mid c_i > t \text{ and } s_i \in [t - c_i + 1 + H, H]\}$ . In particular, any served ship  $i \in Z^{t^{=}}(s) \cup Z^{t^{-}}(s)$  will remain being served at time period t, and the total number of ships that are being processed at a given time period  $t \in T$  is precisely the cardinality of set  $Z^{t^{-}}(s) \cup Z^{t^{-}}(s)$ .

Unfortunately, it is not possible to express this cardinality as a linear expression of the *s* variables. In order to overcome this limitation next we introduce a new set of binary decision variables:

•  $h_{it} \in \{0, 1\}, i \in V, t \in T = \{1, \dots, H\}$ .  $h_{it} = 1 \iff$  service to ship *i* starts at time period *t*.

That is,  $h_{it} = 1 \iff s_i = t$ .

With the aid of variables *h* we can obtain linear expressions for  $|Z^{t^{=}}(s)|$  and  $|Z^{t^{-}}(s)|$ , namely

$$|Z^{t^{=}}(s)| = \sum_{i \in V} \sum_{t'=\max\{1, t-c_i+1\}}^{t} h_{it'} \text{ and } |Z^{t^{-}}(s)| = \sum_{i \in V: \atop c_i > t} \sum_{t'=t-c_i+1+H}^{H} h_{it'}.$$

Hence, the total number of ships being processed at a given time period  $t \in T$  can be written as:

$$\sum_{i \in V} \sum_{t'=\max\{1,t-c_i+1\}}^{t} h_{it'} + \sum_{i \in V: \atop c_i > t} \sum_{t'=t-c_i+1+H}^{H} h_{it'}$$

so the following set of constraints is valid for the SBTP:

$$\sum_{i \in V} \sum_{t'=\max\{1,t-c_i+1\}}^{t} h_{it'} + \sum_{\substack{i \in V:\\c_i>t}} \sum_{t'=t-c_i+1+H}^{H} h_{it'} \le b \qquad t \in T.$$
(9a)

The constraints ensuring that the new variables are well defined and linked to the *s* variables are:

$$\sum_{t \in T} h_{it} = z_{k(i)} \qquad \qquad i \in V \tag{9b}$$

$$s_i = \sum_{t \in T} t h_{it} \qquad i \in V.$$
(9c)

The relationship between the h variables and the existing x variables is quite direct:

$$x_i = \sum_{t=a_i}^{H} h_{it} \qquad i \in V, \tag{9d}$$

whereas in order to relate the *h* variables with the existing *y* variables, we have to observe that when service to a ship  $i \in V$  starts in the cycle next to its arrival cycle (i.e.  $y_i = 1$ ), then its starting time must be some time period smaller than or equal to  $a_i - c_i$ , since otherwise the time between the arrival of the ship and the termination of its service would exceed the duration of the cycle. This means that

$$y_{i} = \begin{cases} \sum_{t=1}^{a_{i}-c_{i}} h_{it} & i \in V \text{ s.t. } a_{i}-c_{i} > 0\\ y_{i} = 0 & i \in V \text{ s.t. } a_{i}-c_{i} \le 0 \end{cases}$$
(12a)

As will be seen in Section 6, where we report numerical results from computational tests, introducing the new set of decision variables h together with the set of constraints (9a)–(12a), has a remarkable effect on the quality of the LP bounds associated with the resulting formulation (see formulation F1 below), which become extremely tight.

F1 min 
$$\sum_{i \in V} w_i + g \sum_{k \in K} C_k (1 - z_k)$$
 (1a)

$$\sum_{i \in V} p_{ij} = z_{k(i)} \qquad \qquad i \in V \tag{4a}$$

$$\sum_{j \in V \setminus \{i\}} p_{ji} \le z_{k(i)} \qquad \qquad i \in V \qquad (4b)$$

$$\sum_{i=V} p_{ii} \le b \tag{4c}$$

$$z_{k(i)} = x_i + y_i \qquad i \in V \qquad (2a)$$
  
$$w_i = s_i - a_i z_{k(i)} + H y_i \qquad i \in V \qquad (2b)$$

$$\begin{aligned} f_i &\leq x_i & i \in V \\ (H+1)f_i &\leq s_i + c_i z_{k(i)} & i \in V. \end{aligned}$$

$$\begin{aligned} a_{i}x_{i} + y_{i} &\leq s_{i} \leq (H - c_{i})x_{i} + c_{i}f_{i} + (a_{i} - c_{i})y_{i} \ i \in V \\ e_{i} &= s_{i} + c_{i}z_{k(i)} - H \ f_{i} \\ s_{i} &\geq e_{j} - H(1 - p_{ij}) \\ w_{i} &\geq e_{j} - a_{i} - H(1 - p_{ij}) \\ \vdots &= s_{i} = V \ i \neq j \ (5c) \\ w_{i} &\geq c_{i} - a_{i} - H(1 - p_{ij}) \\ \vdots &= v \ i \neq j \ (5c) \end{aligned}$$

$$v_i \ge s_i - e_j - H(1 - p_{ij})$$

$$i, j \in V, i \ne j$$

$$(3u)$$

$$o_i \ge o_j + c_j + v_i - H(1 - p_{ij})$$

$$i, j \in V, i \ne j$$

$$(5e)$$

$$v_i < a_i x_i$$

$$i \in V$$

$$(6a)$$

$$o_i \le (H - c_i)(z_{k(i)} - p_{ii}) \qquad i \in V$$
(6b)

$$\sum_{i \in V} \sum_{\substack{t' = \max\{1, t - c_i + 1\}\\H}} n_{it'} + \sum_{\substack{i \in V:\\t - c_i < 0}} t \in T \qquad (9a)$$

$$\sum_{t'=H+(t-c_i+1)}h_{it'}\leq b$$

$$\sum_{t\in T} h_{it} = z_{k(i)} \qquad \qquad i \in V \qquad (\{9b\})$$

$$s_i = \sum_{t \in T} t h_{it} \qquad \qquad i \in V \qquad (\{9c\})$$

$$x_i = \sum_{t=a_i}^{H} h_{it} \qquad \qquad i \in V \qquad (\{9d\})$$

$$y_{i} = \begin{cases} \sum_{t=1}^{a_{i}-c_{i}} h_{it} & i \in V \text{ s.t. } a_{i}-c_{i} > 0\\ y_{i} = 0 & i \in V \text{ s.t. } a_{i}-c_{i} \le 0 \end{cases}$$
(12a)

$$\begin{array}{ll} z_k, x_i, y_i, f_i, h_{it} \in \{0, 1\} \\ s_i, w_i, e_i, v_i, o_i \geq 0 \end{array} & k \in K, i \in V, t(\textcircled{3B}) \\ i \in V. \quad (13b) \end{array}$$

### 3.2. Analysis of formulation F1

The final goal of the SBTP is to determine the ships to be served and to obtain the service sequences to be applied cyclically in each berth. Nevertheless, taking into account that the objective consists of a penalty for each non-served call plus the sum of the waiting times of the served ships, which are dictated by their starting times, the SBTP essentially reduces to identifying the served ships and finding feasible starting times for them. Indeed, feasible starting times can be derived from feasible service sequences. This is, in fact, the main idea in the formulation we have presented, where the starting times of the served ships are determined from the sequences of predecessor variables p and their relation with variables s and w, which is driven by constraints (5b)-(5e). The question that we raise here is whether feasible starting times can be obtained without having an explicit representation of the sequences of ships served in the berths. In particular, whether or not any set of variables z, x, y and s linked via constraints (2a) and (2b), together with a set of variables h satisfying constraints (9a)-(12a)induces a feasible solution to the SBTP.

Regretfully, as shown by the example illustrated in Fig. 2, the answer to the above question is negative, indicating that constraints (9a)–(12a) are not sufficient to guarantee that a feasible SBTP solution can be obtained. The example considers a cycle duration H = 8 and  $V = \{1, 2, 3\}$ , where the processing time of all three ships is five units ( $c_i = 5$  for all  $i \in V$ ). It is easy to check that if the number of available berths is b = 2, there is no feasible solution where all three ships are served (independently of what the arrival times for the ships are). However, as Fig. 2 shows it is possible to find starting times for the ships that satisfy constraints (9a), i.e. starting times such that at each time period at most two ships are being processed. In the solution depicted in the figure  $s_1 = 1$ ,  $s_2 = 3$ , and  $s_3 = 6$ . That is,  $h_{1t} = 1$  for all  $t \in [1, 5]$ ;  $h_{2t} = 1$  for all  $t \in [3, 7]$ ; and,  $h_{3t} = 1$  for all  $t \in [1, 2] \cup [6, 8]$ . As can be seen, these values satisfy constraints (9a).

Therefore, we conclude that the formulation F2 below is a relaxation of the SBTP:

F2 min 
$$\sum_{i \in V} w_i + g \sum_{k \in K} C_k (1 - z_k)$$
 (1a)

$$z_{k(i)} = x_i + y_i \qquad \qquad i \in V \qquad (2a)$$

$$w_i = s_i - a_i z_{k(i)} + H y_i \qquad i \in V$$
(2b)

$$\sum_{i \in V} \sum_{t'=\max\{1,t-c_i+1\}} h_{it'} + \sum_{\substack{i \in V: \\ t-c_i < 0}} t \in T$$
(9a)

$$\sum_{t'=H+(t-c_i+1)} h_{it'} \le b$$

$$\sum_{t \in T} h_{it} = z_{k(i)} \qquad \qquad i \in V \qquad (\{9b\})$$

$$s_i = \sum_{t \in T} t h_{it} \qquad i \in V \qquad (\{9c\})$$

$$x_i = \sum_{t=a_i}^{H} h_{it} \qquad \qquad i \in V \qquad (\{9d\})$$

$$y_{i} = \begin{cases} \sum_{t=1}^{a_{i}-c_{i}} h_{it} & i \in V \text{ s.t. } a_{i}-c_{i} > 0\\ y_{i} = 0 & i \in V \text{ s.t. } a_{i}-c_{i} \le 0 \end{cases}$$
(12a)

$$\begin{aligned} & z_k, x_i, y_i, h_{it} \in \{0, 1\} \\ & s_i, w_i \ge 0 \end{aligned} \qquad k \in K, i \in V, t \in T \quad (14a) \\ & i \in V. \quad (14b) \end{aligned}$$



**Fig. 2.** Example with H = 8,  $V = \{1, 2, 3\}$ ,  $c_i = 5$ ,  $i \in V$ , and b = 2 where no feasible solution serving all ships exists, but values  $h_{it}$  satisfying constraints (9a) can be found.

The relationship between the optimal values of *F*2 and *SBTP* is summarized below, where  $v(\cdot)$  denotes the optimal value of a given optimization problem.

### **Proposition 1.** $v(F2) \leq v(SBTP)$ .

Since a feasible solution to formulation F2 does not necessarily induce a feasible solution to the SBTP, we now address the question of whether we can know if there is a feasible solution to the SBTP supported by a given vector  $(\bar{z}, \bar{h})$  in the feasible domain of F2. As we next explain, the answer to this question can be obtained by solving an auxiliary problem, that in the following will be referred to as  $AP(\bar{z}, \bar{h})$ , which can be used as an oracle. Problem  $AP(\bar{z}, \bar{h})$  assumes that all the ships indexed in  $\overline{V} = \{i \in V : \bar{z}_{k(i)} = 1\}$ must be served and their service starting times are those dictated by h. Essentially,  $AP(\bar{z}, h)$  rephrases the above question in terms of finding an assignment to berths of the ships indexed in  $\overline{V}$ , such that the overall service time of all the ships assigned to the same berth does not exceed the duration of a cycle, and minimizes the overall service overlap at berths. Since ideally each berth has a service capacity of one at each time period, we define the service overlap at a berth at a given time period t as the excess of ships allocated to the berth at time period t. This excess is given by the number of ships allocated to the berth being served at time period t minus one, or zero when this quantity is negative.

In the following, let  $R = \{1, ..., b\}$  denote the index set for the berths, and  $\overline{V}^t = \{i \in \overline{V} : \overline{h}_i \le t \text{ and } \overline{h}_i + c_i - 1 \ge t\} \cup \{i \in \overline{V} : \overline{h}_i > t \text{ and } (\overline{h}_i + c_i - 1) \ge t + H\}$  the set of ships indexed in  $\overline{V}$  being served at time period *t* assuming that their service starting times are dictated by  $\overline{h}$ .

For each  $i \in \overline{V}$ ,  $r \in R$ , let  $\lambda_{ir} \in \{0, 1\}$ , be a binary variable, which takes the value 1 if and only if ship *i* is allocated to berth *r*. Associated with each berth  $r \in R$  and time period  $t \in T$  let us consider another decision variable  $\sigma_{rt}$  indicating the *service overlap* in berth *r* at time period *t*. That is,  $\sigma_{rt} = \max\{\sum_{i\in\overline{Z}^t} \lambda_{ir} - 1, 0\}$  is the excess relative to the service capacity of berth *r* at time period *t*.

The auxiliary allocation problem that we consider is therefore:

$$AP(\bar{z}, \bar{h}): \min \sum_{r \in \mathbb{R}} \sum_{t \in T} \sigma_{rt}$$
 (15a)

$$\sum_{r \in \mathbb{R}} \lambda_{ir} = 1 \qquad \qquad i \in \overline{V} \tag{15b}$$

$$\sum_{i\in\overline{V}}c_i\lambda_{ir}\leq H \qquad r\in R \qquad (15c)$$

$$\sigma_{rt} \ge \sum_{i \in V^t} \lambda_{ir} - 1 \qquad r \in R, t \in T$$
(15d)

$$\lambda_{ir} \in \{0, 1\} \qquad i \in \overline{V}, r \in R \qquad (15e)$$
  
$$\sigma_{rt} \ge 0 \qquad r \in R, t \in T. \qquad (15f)$$

Constraints (15b) guarantee that all the ships of  $\overline{V}$  are allocated to some berth and (15c) that the total service time of all the ships allocated to the same berth does not exceed the cycle duration. Finally, Constraints (15d) determine the overlaps, whose total amount is minimized.

Note that  $AP(\bar{z}, \bar{h})$  is a variation of a bin packing problem (see, e.g. chapter 18 in Korte & Vygen, 2006), where the capacity of the bins is H and the demand of item  $i \in \overline{V}$  is  $c_i$ , and recall that bin packing is known to be NP-hard (Garey & Johnson, 1979). Moreover, since we impose that all the ships indexed in  $\overline{V}$  are allocated, it may happen that its feasible domain is empty. Let  $\Omega_{AP(\bar{z},\bar{h})} = \{(\lambda, \sigma) \in \{0, 1\}^{|\overline{V}|} \times \mathbb{R}^+: \text{satisfying (15b)-(15f)}\}$  denote the feasible domain of  $AP(\bar{z}, \bar{h})$ .

### **Proposition 2.**

- If  $\Omega_{AP(\overline{z},\overline{h})} = \emptyset$ , then there is no feasible solution to SBTP serving all the ships indexed in  $\overline{V}$ .
- Suppose  $\Omega_{AP(\bar{z},\bar{h})} \neq \emptyset$ . Then,
  - There is a feasible solution to the SBTP that serves all the ships indexed in  $\overline{V} = \{i \in V : \overline{z}_{k(i)} = 1\}$  with starting times given by  $\{\overline{h}_{it}\}_{i \in \overline{V}, t \in T}$  if and only if  $v(AP(\overline{z}, \overline{h}))$  is zero.
  - Suppose  $v(AP(\bar{z}, \bar{h})) = 0$ , and let  $\bar{\lambda}$  be the allocation vector associated with an optimal solution to  $AP(\bar{z}, \bar{h})$ . Then the solution  $(\bar{z}, \bar{h})$  where each ship  $i \in \overline{V}$  starts its service at time period t with  $\bar{h}_{it} = 1$  in the berth  $r \in R$  such that  $\bar{\lambda}_{ir} = 1$  is an optimal SBTP solution.

When  $\Omega_{AP(\bar{z},\bar{h})} \neq \emptyset$ , but  $\nu(AP(\bar{z},\bar{h})) > 0$ , then no feasible solution to SBTP exists serving all the ships of  $\overline{V}$  with starting times  $\{\bar{h}_{it}\}_{i\in\overline{V},t\in T}$ . Still, from an optimal allocation to  $AP(\bar{z},\bar{h})$  a feasible SBTP solution can be obtained heuristically. Since the total service time of all the ships allocated to the same berth does not exceed the cycle duration, it is possible to sequence all these ships in such a way that there are no service overlaps, although this will carry changes in the service starting times of some ships and, in its turn, in their waiting times, as will be discussed in Section 5.

### 4. An SBTP formulation with disaggregated service time variables

In this section we introduce our final formulation for the SBTP, based on the idea of counting the number of ships that are served simultaneously at a given time period that overcomes the difficulties discussed in the previous section, basically derived from the fact that constraints (9a) aggregate the service occupation of all the berths. Hence, what we propose is to redefine the discretized binary variables  $h_{it}$ ,  $i \in V$ ,  $t \in T$ , in such a way that the berth to

Table 1					
Number of variables	and	constraints	in	the	formulations.

Formulation	Variables		Constraints
	Binary	Continuous	
FO	p, z, x, y, f  V ( V  - 1) + 4 V	s, w, e, v, o 5 V	4 V ( V -1) + 10 V  + 1
F1	p, z, x, y, f, h $ V ( V  - 1) +  V  \times  T  + 4 V $	s, w, e, v, o 5 V	4 V ( V -1) + 14 V  +  T  + 1
F2	$z, x, y, h$ $ V  \times  T  + 3 V $	s, w 2 V	6 V  +  T
F3	$z, x, y, \hat{h}$ $3 V  +  V  \times  T  \times  R $	s, w 2 V	$6 V  +  T  \times  R $

which each served ship is allocated is made explicit. That is, consider the set of decision variables  $\hat{h}_{itr} \in \{0, 1\}$ ,  $i \in V$ ,  $t \in T$ ,  $r \in R$ , such that

 $\hat{h}_{itr} = 1 \iff$  service to ship i starts at time period t in berth r.

Now the constraints that guarantee that at most one ship is being served at each berth at each time period are

$$\sum_{i \in V} \sum_{t'=\max\{1,t-c_i+1\}}^{t} \hat{h}_{it'r} + \sum_{\substack{i \in V: \\ c_i > t}} \sum_{t'=t-c_i+1+H}^{H} \hat{h}_{it'r} \le 1 \qquad r \in R, \ t \in T.$$
(16a)

Given that constraints (16a) prevent service overlaps within the same berth, neither variables  $e_i$ ,  $v_i$ , and  $o_i$ , nor constraints (5a)–(5e), (6a) and (6b) are needed anymore, since their role was to prevent such infeasibilities. Therefore, taking into account that the relation  $h_{it} = \sum_{r \in R} \hat{h}_{itr}$ , for all  $i \in V$ ,  $t \in T$ , we have the following valid formulation for the SBTP:

F3 min 
$$\sum_{i \in V} w_i + g \sum_{k \in K} C_k (1 - z_k)$$
 (1a)

$$z_{k(i)} = x_i + y_i \qquad i \in V$$

$$w_i = s_i - q_i z_i + H y_i \qquad i \in V$$
(2a)
(2b)

$$\begin{cases} \sum_{i=1}^{a_i-c_i} \sum_{i=1}^{a_i-c_i} \hat{h}_{i}, & i \in V \text{ st. } a_i-c_i > 0 \\ \sum_{i=1}^{a_i-c_i} \sum_{i=1}^{a_i-c_i} \hat{h}_{i}, & i \in V \text{ st. } a_i-c_i > 0 \end{cases}$$

$$y_{i} = \begin{cases} \sum_{t=1}^{a_{i} - c_{i}} \sum_{r \in R} n_{itr} & i \in V \text{ s.t. } a_{i} - c_{i} > 0\\ y_{i} = 0 & i \in V \text{ s.t. } a_{i} - c_{i} \le 0 \end{cases}$$
(12a)

$$\sum_{i \in V} \sum_{t'=\max\{1, t-c_i+1\}} \hat{h}_{it'r} \\ + \sum_{i \in V: t'=H+(t-c_i+1)} \prod_{t' \in V} \hat{h}_{it'r} \le 1$$

$$(16a)$$

$$\sum_{t \in T} \sum_{r \in R} \hat{h}_{it} = z_{k(i)} \qquad \qquad i \in V$$
(17a)

$$s_i = \sum_{t \in T} \sum_{r \in R} t \ \hat{h}_{itr} \qquad i \in V,$$
(17b)

$$x_i = \sum_{t=0}^{H} \sum_{r \in \mathbb{R}} \hat{h}_{itr} \qquad i \in V,$$
(17c)

$$z_{k}, x_{i}, y_{i}, \hat{h}_{itr} \in \{0, 1\} \qquad k \in K, i \in V, t \in T, r \in R$$
(17d)  
$$s_{i}, w_{i} > 0 \qquad i \in V.$$
(17e)

### 4.1. Comparison of formulations

Table 1 summarizes a theoretical comparison of the formulations we have introduced, based on the number and type of variables and constraints that they involve. Observe that even if variables  $s_i$ ,  $w_i$ ,  $e_i$ ,  $v_i$ ,  $o_i$  are restricted to take integer values and thus should be defined as integer, they can be relaxed to take nonnegative values. The reason is that the constraints relate them to the *x*, *y* and *f* variables, which are restricted to take binary values, guarantee that those variables will take integer values in any optimal solution. This information, will be complemented in Section 6 with an empirical comparison of the performance of the formulations, based on the numerical results produced by each of them in the computational tests that we have carried out.

We close this section by pointing out that even if the formulations we have introduced produce SBTP solutions, the obtained solutions may be not sufficiently explicit, in the sense that in some cases they do not give the specific allocations of served ships to berths, or they do not obtain the specific sequences of consecutive ships served in each berth. Such issues as well as other related ones, become of interest when dealing algorithmically with the SBTP and will be addressed in the next section, where we focus on how to fully determine feasible solutions for the SBTP from partial or infeasible information provided by the proposed formulations.

### 5. Algorithmic issues for fully determining feasible SBTP solutions

A feasible SBTP solution is fully determined by (*i*) the set of served ships, (*ii*) the service time of each ship, and (*iii*) the allocation of served ships to berths. All the formulations that we have introduced include explicit information on items (*i*) and (*ii*), via decision variables  $z_k$ ,  $k \in K$  and  $s_i$ ,  $i \in V$ , respectively, which are the two sets of decision variables common to all four formulations. Still, except for formulation F3, in which the expression  $\sum_{t \in T} \hat{h}_{irt}$  gives explicit information on whether or not ship  $i \in V$  is allocated to berth  $r \in R$ , all other formulations omit such information. Moreover, even if starting times together with the explicit allocation of ships to berths determine the service sequence of each berth, this information is not explicit in any of the formulations F2 and F3 because they do not include explicit sequencing information.

On the other hand, in Section 3.2 we have seen that F2 is a relaxation that does not necessarily produce feasible SBTP solutions, although in some cases the auxiliary subproblems  $AP(\bar{z}, \bar{h})$ ,  $r \in R$  give assignments of ships to berths that may result in feasible solutions. Obtaining feasible solutions from the information provided by these auxiliary subproblems can be useful, not only within an algorithmic framework based on F2, but also to *support* any valid formulation with some initial feasible solution, which could be used as an incumbent and could possibly reduce the computational burden needed to optimally solve the formulations.

In the remainder of this section first we give simple procedures to obtain explicit allocations to berths and to obtain explicit sequences of consecutive ships served in each berth and discuss some related algorithmic issues. Then we give a simple template for finding feasible SBTP solutions.

# 5.1. Procedure for determining explicitly the set of ships served in the same berth from a pair of vectors (z, p) satisfying (4a)-(4c)

From a vector  $\bar{z}$  representing the set of served ships and a predecessors vector  $\bar{p}$  satisfying (4a)–(4c), the set of ships that are allocated to each berth  $r \in R$ ,  $B^r$ , can be easily identified by tracing back the predecessor variables. For this we perform |R| iterations where at each of them we select a berth index  $r \in R$  not yet considered and use the predecessors vector to determine the set of ships  $B^r$  that will be allocated to it, using the following two steps; (i) identify a served ship  $\hat{i}$ , not yet allocated, that does not precede any other served ship (i.e.  $\hat{i}$  such that  $\bar{z}_{k(\hat{i})} = 1$  and  $\sum_{j \in V \setminus \{\hat{i}\}} \bar{p}_{j\hat{i}} = 0$ ); and, (ii) trace back the sequence of ships that precede i using the predecessors vector p, by iteratively identifying the only index  $\hat{j} \in V$  that precedes the current ship  $\hat{i}$  (i.e. the only  $\hat{j}$  s.t.  $\bar{p}_{\hat{i}\hat{j}} = 1$ ) and updating the index of the curent ship î to that of its predecessor (i.e.  $\hat{i} \leftarrow \hat{j}$ ). Step (*ii*) is repeated until  $\hat{i}$  is the first ship procesed in the berth (i.e.,  $\hat{j} = \hat{i}$ ). Note that, given that in the SBTP all the berths have the same characteristics, the obtained ship subsets  $B^r$ are perfectly interchangeable among them.

## 5.2. Procedure for determining explicitly the predecessors vector associated with a feasible solution to F3

It is clear that any feasible solution to F0 or F1, with starting times dictated by a given vector  $\bar{s}$ , defines a feasible solution to F3, with  $\hat{h}_{itr} = 1$  if and only if  $\bar{s}_i = t$  and  $i \in B^r$ , where the sets of ships served in the same berth,  $\{B^r\}_{r \in R}$ , can be identified with an algorithm based on the description of Section 5.1. Reciprocally, from a feasible vector  $\hat{h}^*$  in the domain of F3 we can obtain a feasible solution to F0 or F1 by defining the predecessors vector  $\hat{p}$  induced by the solution  $\hat{h}$  with a procedure that initially determines the set of ships served in each berth, i.e.  $B^r = \{i \in V : \sum_{t \in T} \hat{h}_{irt}^* = 1\}, r \in R$ , and then traces forward the sequence of ships indexed in each  $B^r$  by determining the non-zero components of vector  $\hat{h}^*$  with r fixed for progressively increasing time indices  $t \in T$ . Details are omitted.

### 5.3. Finding feasible berth allocations from scratch

Feasible allocations of ships to berths can also be obtained from scratch, without information on the set of served ships or the predecessors vector produced by formulations F0 or F1. For this, we can solve a variation of the auxiliary problem  $AP(\bar{z}, \bar{h})$  in which the set of ships to be served is not fixed in advance (i.e. z is an additional set of decision variables so there is no parameter  $\bar{z}$ ) and the arrival times are used as tentative service times (i.e.  $\bar{h}_{it} = 1$  if an only if  $t = a_i$ ,  $i \in V$ ). Since now the set of ships to be served is not known in advance, in order to guarantee mother-ship restrictions, Constraints (15b) are stated as:

$$\sum_{r \in \mathcal{R}} \lambda_{ir} = z_{k(i)}, \qquad i \in V.$$
(18a)

Together with Constraints (15c), the above constraints (18a) ensure that a feasible allocation of served ships to berths is obtained. Constraints (15d) play the same role as in the original  $AP(\bar{z}, \bar{h})$  formulation and identify the overlaps that are produced. Now, in order to attain a tradeoff between served ships (which, as said, are not known in advance) and service overlap (which can be quite high, given that service times are set to arrival times) a suitable objective is to consider a weighted combination of both criteria by maximizing:

$$\mu \sum_{i \in V} c_i \sum_{r \in R} \lambda_{ir} - \sum_{r \in R} \sum_{t \in T} \sigma_{rt},$$
(19a)

where  $\mu$  is a parameter balancing the two terms of the involved objective function.

The above problem will be referred to as  $\overline{AP}(\overline{h})$  where, as indicated in the beginning of this section, the vector  $\overline{h}$  is dictated by the arrival times. The set of ships served in each berth can be easily identified from an optimal solution to  $\overline{AP}(\overline{h})$ ,  $(\hat{z}, \hat{\lambda}, \hat{\sigma})$ . In particular,  $B^r = \{i \in V : \hat{\lambda}_{ir} = 1\}, r \in R$ .

## 5.4. Determining a service sequencing for the set of ships allocated to the same berth

When the set of ships  $B^r$  allocated to berth  $r \in R$  is given, an optimal service sequence for the ships of  $B^r$  that minimizes the total waiting times can be obtained with a formulation that particularizes F2 for just one berth. Such a formulation assumes that all the ships indexed in  $B^r$  can be served within one cycle, i.e. it is assumed that the condition  $\sum_{i \in B^r} c_i \leq H$  holds. Since we are restricting to one single berth, no service overlaps may appear, meaning that the right hand side of the updated constraint (9a) must be 1. The formulation corresponding to berth r is therefore:

$$SEQ_r \min \sum_{i \in \mathbb{R}^r} w_i \tag{20a}$$

$$x_i + y_i = 1 \qquad \qquad i \in B^r$$
 (20b)

$$w_i = s_i + H y_i - a_i \qquad i \in B^r$$
(20c)

$$\sum_{i \in V} \sum_{\substack{t' = \max\{1, t - c_i + 1\} \\ H}} h_{it'} \\ + \sum_{\substack{i \in B': \\ t - c_i < 0}} \sum_{t' = H + (t - c_i + 1)} h_{it'} \le 1$$
(20d)

$$\sum_{t \in T} h_{it} = 1 \qquad i \in B^r$$
 (20e)

$$s_i = \sum_{t \in T} t \ h_{it} \qquad i \in B^r$$
 (20f)

$$x_i = \sum_{t=a_i}^{H} h_{it} \qquad \qquad i \in B^r$$
 (20g)

$$y_i = 0 \qquad i \in B^r \text{ s.t. } a_i - c_i \le 0 \quad (20h)$$

$$y_i = \sum_{t=1}^{a_i - c_i} h_{it}$$
  $i \in B^r$  s.t.  $a_i - c_i > 0$  (20i)

$$\begin{array}{ll} x_i, y_i, h_{it} \in \{0, 1\} & i \in B^r, t \in T \\ s_i, w_i \ge 0 & i \in B^r. \end{array}$$
 (20j)

Observe that the waiting times resulting from arrival times  $a_i$ , can be seen as the *tardiness* relative to the due dates  $a_i + c_i - 1$ . Therefore SEQ<sub>r</sub> is an exact formulation for the minimization of the total tardiness as defined above. As shown in Du & Leung (1990) this problem is already NP-hard.

### 5.5. Algorithmic framework for finding feasible SBTP solutions

Based on the ingredients described in the previous sections, an algorithmic template for building feasible SBTP solutions is to apply the following three steps:

- S1 Determine a subset of served ships,  $\overline{V}$ , and tentative service times for the selected ships,  $\overline{h}_i$ ,  $i \in \overline{V}$ , respecting the mothership constraints (2a).
- S2 Determine an allocation of the ships of  $\overline{V}$  to berths,  $\{B^r\}_{r \in \mathbb{R}}$ , that satisfies the cycle duration, i.e.  $\sum_{i \in B^r} c_i \leq H$ ,  $r \in \mathbb{R}$ , deviating as little as possible from the tentative service times  $\overline{h}_i$ ,  $i \in \overline{V}$ .
- S3 For each berth  $r \in R$ , build a service sequence for the ships allocated to it,  $\{B^r\}$ .

There are multiple ways in which the above algorithmic scheme can be implemented. Below we outline the two alternatives that we have implemented and tested in our computational experiments. Both alternatives differ in how steps S1 and S2 are defined, but, in essence, share step S3, which consists in solving the subproblems SEQ<sub>r</sub> induced by the subsets  $\{B^r\}_{r \in \mathbb{R}}$  produced by step S2 (see Section 5.4).

<u>HEUR</u>: (simple heuristic based on the solution of subproblem  $\overline{AP}(\overline{h})$  described in Section 5.3). In this heuristic steps S1 and S2 are merged in one single step, in which  $\overline{AP}(\overline{h})$  is solved using arrival times as service times, and Constraints (18a) instead of Constraints (15b).

- S1 + S2: Solve  $\overline{AP}(\overline{h})$ .
- S3: Solve the subproblems SEQ<sub>r</sub> induced by the subsets  $\{B^r\}_{r \in R}$  obtained in S1+S2.

<u>2-phase solution algorithm</u>. Recall that Proposition 2 gives us a simple check for the optimality of the solutions produced by F2. In particular, when  $v(AP(\bar{z}, \bar{h})) = 0$ , the solution produced by F2 is feasible for SBTP and thus optimal. This naturally leads to a 2-phase solution algorithm in which the first phase is step S1 when formulation F2 is solved. In the second phase, which is determined by S2+S3, the solution of the auxiliary problem  $AP(\bar{z}, \bar{h})$  is followed by the solution of the subproblems SEQ<sub>r</sub>, associated with the resulting sets of ships  $B^r$ ,  $r \in R$ . In particular, the procedure is as follows:

S1 Solve formulation F2.

- S2 Solve the auxiliary problem  $AP(\bar{z}, \bar{h})$ .
- S3 Apply the Feasibility Check based on Proposition 2.
- If  $(v(AP(\bar{z}, \bar{h})) = 0)$  then (the solution obtained in S2 is optimal)
  - Else (the current solution is not optimal)
  - Solve the subproblems SEQ<sub>r</sub> induced by the subsets  $\{B^r\}_{r\in R}$  obtained in S2.

Observe that Step S1 of the 2-phase algorithm produces a valid lower bound, which is not necessarily associated with a feasible SBTP solution, whereas the second phase produces a valid upper bound associated with the feasible SBTP solution obtained after applying S2 + S3.

### 6. Computational experiments

In order to study the empirical performance of the formulations introduced in the previous sections we have carried out a series of computational experiments whose numerical results are presented and analyzed in this section.

All the computational tests have been carried out in an DELL XPS 15 9550 Intel i7-6700HQ 2.6 GHz with 16 GB RAM, under Windows 10 Pro as operating system. All formulations have been coded in Mosel 5.2.0 with Xpress Optimizer Version 36.01.03 using as solver (Xpress, 2020).

For the experiments we have used the set of 96 SBTP benchmark instances generated by the authors of Iris et al. (2018) based on a prototypical instance of Imai et al. (2014), that they used in their computational experiments. These instances are classified according to their number of calling ships  $n \in \{50, 70, 100, 150\}$ , number of berths  $b \in \{4, 8, 12\}$ , as well as the following characteristics:

- Proportion of small (feeder; F), medium (M) and large (jumbo; J) calling ships, which are determined by their service times (in hours). In instances labeled *E* ("equal") the proportion of ships of types F, M and J is 33.3%, 33.3%, and 33.4% respectively, whereas instances labeled *A* ("alternative") the proportions are 60%, 30%, and 10%, respectively.
- Service times, which represent handling times (in hours). They have been generated from integer uniform distributions which,

on the one hand, depend on the ships characteristics (F, M, or J), and, on the other hand, can be either S ("small") or L ("large").

- Service times for instances labeled S are drawn from U[4, 8], U[6, 10] and U[8, 12] for ships of type F, M and J, respectively.
- Service times for instances labeled L are drawn from *U*[8, 10], *U*[10, 14] and *U*[14, 22] for ships of type F, M and J, respectively.

All instances are available at https://github.com/elalla/ strategic-berth-template-problem. Each instance is identified with a label "i\_n\_b\_c\_s", where *i* is the numeric label assigned to the instance in Iris et al. (2018),  $n \in \{50, 70, 100, 150\}$  its number of ships,  $b \in \{4, 8, 12\}$  the number of berths,  $c \in \{A, E\}$  the type of composition, and  $s \in \{S, L\}$  the type of service. In all instances the planning horizon has a duration of 152 hours which is a prototype week that will be repeated cyclically. The number of connections between mother/feeder ships depends on the instance size, although usually 10 - 20% of all ships are in a mother-feeder link. Finally, the penalty for each rejected call is g = 10,000 (see Iris et al., 2018 for further details).

As we next explain this very high value of the penalty g plays an essential role for determining the optimal values of the considered SBTP instances. For this, we observe that, in the objective function, the overall penalty associated with non-served calls,  $g \sum_{k \in K} C_k (1 - z_k)$ , fully dominates the term  $\sum_{i \in V} w_i$  associated with the waiting times. Since the maximum waiting time of any served ship is  $w_i \leq H$ , a very crude upper bound of  $\sum_{i \in V} w_i$  is  $n \times H$ , which for our considered parameter values  $n \le 150$  and H = 152indicates that for any of the considered instances it holds that  $\sum_{i \in V} w_i \le 150 \times 152 = 22,800$ . On the other hand, since  $c_i \ge 4$  for all  $i \in V$  (the value of 4 corresponds to the lower limit in U[4, 8]for instances labeled S of type F), the coefficient  $C_k = \sum_{i \in V_k} c_i \ge 4$ , for all  $k \in K$ , so each non-served class  $k \in K$  contributes to the objective function with a penalty greater than or equal to  $g \times C_k \ge$  $4 \times 10,000 = 40,000$ . This means that the individual penalty corresponding to each non-served class exceeds the maximum possible total value of the term  $\sum_{i \in V} w_i$ . Thus any optimal solution will reject as few service calls as possible or, equivalently, any optimal solution will serve as many classes as allowed by the overall berths capacity. Indeed it is possible to determine a priori the optimal value for  $g \sum_{k \in K} C_k (1 - z_k)$  by finding a subset of ship classes that can be assigned to the available berths with no overlaps (ignoring any sequencing issues) of maximum value for  $\sum_{k \in K} C_k z_k$ . Such a set can be found by solving the problem

$$S^* = \max \sum_{k \in K} C_k Z_k \tag{21a}$$

$$\sum_{r \in \mathbb{R}} \lambda_{ir} \le z_{k(i)} \qquad \qquad i \in \overline{V}$$
(21b)

$$\sum_{i \in \overline{V}} c_i \lambda_{ir} \le H \qquad r \in R \tag{21c}$$

$$\lambda_{ir} \in \{0, 1\} \qquad \qquad i \in \overline{V}, r \in R \qquad (21d)$$

$$z_k \in \{0, 1\}. \qquad \qquad k \in K \qquad (21e)$$

Thus, if  $S^*$  is the maximum capability of a given instance in terms of the overall service time of the accepted ships then, for any optimal solution to the instance, the overall service time of the accepted ships will be precisely  $S^*$ , That is, the total service time of rejected ships will be  $\sum_{j \in V} c_j - S^*$ , so the overall penalty for the rejected ships is a constant  $P^* = g \times (\sum_{j \in V} c_j - S^*)$ . Note finally that the value of the maximum service capability of a given instance,  $S^*$ , can be computed by solving a variation of a bin packing problem (see, e.g. chapter 8 in Martello & Toth, 1990) in which there are *b* bins each of them with capacity *H* and the demand of

European Journal of Operational Research 298 (2022) 99-117

 Table 2

 Summary of instances characteristics.

b	$B = H \times b$	n	L/S	A/E	Instances	$D = \sum_{i \in V} c_i$		R = D/B	P*
4	608	50	S	А	1-4	341	50	0.56	0
				Е	5-8	399	75	0.66	0
			L	А	13-16	551	00	0.91	0
				Е	9-12	634	50	1.04	265,000
		70	S	А	71-20	464	00	0.76	0
				Е	21-24	559	25	0.92	0
			L	Α	29-32	744 75		1.22	1,367,500
				E	25-28	864 25		1.42	2,562,500
		100	S	А	33-36	676	25	1.11	682,500
				E	37-40	810	50	1.33	2,025,000
			L	А	45-48	1067	50	1.76	4,595,000
				Е	41-44	1195	25	1.97	5,872,500
8	1216	70	S	А	61-64	396	00	0.33	0
				Е	57-60	487	25	0.40	0
			L	Α	49-52	784	50	0.65	0
				Е	53-56	919	50	0.76	0
		100	S	Α	65-68	581	50	0.48	0
				Е	77-80	700	25	0.58	0
			L	Α	69-72	1016	25	0.84	0
				Е	73–76	1234	75	1.02	202,500
12	1824	150	S	А	93-96	865	25	0.47	0
				Е	81-84	1054	00	0.58	0
			L	Α	89-92	1492	75	0.82	0
				Е	85-88	1752	75	0.96	0

the items is  $c_i$ ,  $i \in V$ , where the condition that all the ships in each class are either allocated or rejected is guaranteed by the right hand side of (21b). For a better assessment of the results of our computational experiments, for each instance, we have precomputed the value of its penalty  $P^*$ .

Table 2 summarizes the instances characteristics. The first three columns indicate the number of berths, *b*, the overall *service capability*  $B = b \times H$ , and the number *n* of calling ships respectively. The following two columns indicate whether service times are *S/L* and whether the proportion of small/large/jumbo calling ships is *A/E*, respectively. Column labeled *Instances* indicates the range of numeric labels of the four benchmark instances with those specific parameters, whereas columns under  $D = \sum_{i \in V} c_i$  and R = D/B give the averages, over those four instances, of the overall service demand *D* and the *demand rate*, *R*, respectively. The last column of Table 2 gives average values of  $P^*$  for the different instance classes, precomputed as explained above.

In the following the set consisting of the four instances  $*\_n\_b\_c\_s$  sharing the same parameter values for *n*, *c*, *b* and *s* will be denoted by  $C_{n\_b\_c\_s}$ . In Table 2 and the rest of this section, results are presented for the different groups which, for the same number of berths and calling ships, are ordered by increasing value of demand rate. Note that this grouping does not correspond to increasing values of the numerical labels of the instances, which we always indicate so the specific instances to which the results correspond can be identified.

### 6.1. Numerical results for instances with four berths

We start our analysis by comparing several formulations and simple algorithmic schemes among them on the instances with four berths, all of which have a number of calling ships  $n \in \{50, 70\}$ . That is, we consider the classes  $C_{n\_4\_c\_s}$ , for varying values of the remaining parameters.

Preliminary testing indicated that formulation F0 produced very weak LP bounds and was only able to solve to proven optimality small size instances. We thus excluded it from further consideration and focused on the remaining formulations. In the remainder of this section we will compare the following bounds:

• Lower bounds:

- $L_0$ : value of the LP relaxation of formulation F1.
- L<sub>1</sub>: lower bound produced by formulation F1 at termination.
   L<sub>2</sub>: optimal value of formulation F2
- (valid lower bound, obtained in S2 of the 2-phase algorithm).
- L<sub>3</sub>: lower bound produced by formulation F3 at termination.Upper bounds:
  - $U_0$ : value of the feasible solution obtained with the heuristic *HEUR* described in Section 5.5. In the objective function (19a) of problem  $\overline{AP}(\bar{h})$ , the weight that has been used for the combination of the *service* and *overlap* criteria is  $\mu = 100$ . This value was chosen after some preliminary testing, where we observed that this is a good tradeoff of the two terms of the involved objective function.
  - *U*<sub>1</sub>: upper bound corresponding to the value of the best solution produced by formulation F1 at termination.
  - $U_2$ : upper bound associated with the solution produced by the 2-phase algorithm of described in Section 5.5
  - *U*<sub>3</sub>: upper bound corresponding to the value of the best solution produced by formulation F3 at termination.

F3 produced proven optimal solutions for all the considered instances within a maximum computing time of 3600 seconds, with the exception of instance 21\_70\_4B\_E\_S, for which F3 consumed over 9000 seconds. This allows us to report percentage deviations of the above lower and upper bounds relative to optimal values, as well as percentage optimality gaps of the intervals  $[L_i, U_i]$ , i = 0, 1, 2, 3. In all cases, the bounds that we analyze correspond to waiting times. That is, if the value of a solution is  $W + P^*$ , where  $W = \sum_{i \in \overline{V}} w_i$  is the total waiting time of the served ships, we exclude the constant penalty term P\*, and only consider the value W. Otherwise, relatively large differences in the value of W may somehow be hidden behind the large value of the penalty term  $P^*$ . Therefore, the actual bounds that we consider for i = 0, ..., 3 are  $L_i^w$  and  $U_i^w$ , such that  $L_i = L_i^w + P^*$  and  $U_i = U_i^w + P^*$ . Then, if  $W^*$  is the overall waiting time in an optimal solution, the percentage deviations and optimality gaps that we report are defined as follows:

• Deviations of the lower bounds from optimal values:  $%DL_i^w = 100(W^* - L_i^w)/W^*$ , i = 0, ..., 3.

Summary of numerica	l results for	small instances	with $b = 4$	and $n \in \{50, 70\}$	)}.
---------------------	---------------	-----------------	--------------	------------------------	-----

n:		50				70						
L/S-A/E	:	S-A	S-E	L-A	L-E	S-A	S-E	L-A	L-E			
Inst. lal	oels:	1-4	5–8	13–16	9–12	17–20	21–24	29-32	25-28			
HEUR	$DL_0^w$	0.00	0.10	1.42	4.66	1.13	0.91	8.04	7.85			
	$DU_0^w$	31.40 38.28 46.46 92.77		59.04	43.40	246.14	179.85					
	$%G_0^w$	31.40	38.42	48.52	103.92	61.00	44.79	281.38	203.05			
	CPU <sub>0</sub>	0.74 0.69 3.59 12.18		26.80	4.29	7.30	6.03					
	#Opt <sub>0</sub>	0 0 0 0		0	0	0	0	0				
F1	$DL_1^w$	0.00 0.00 1.49 0.11		0.11	0.00	0.17	4.86	4.44				
	$DU_1^w$	0.00	0.00	76.55	43.60	11.86	33.79	126.25	125.64			
	$%G_1^{\dot{w}}$	0.00	0.00	80.01	43.25	11.86	30.00	137.52	136.52			
	$CP\dot{U}_1$	2.54	6.71	3599.81	3600.29	1846.31	3599.87	3599.86	3600.51			
	$#Opt_1$	4	4	0	0	2	0	0	0			
F2	$DL_2^w$	0.00	0.00	0.00	0.00	0.00	2.82					
	$TL_2^{\overline{w}}$	0.21	0.26	2.56	7.83	0.79	6.76	19.35	15.74			
	$DU_2^w$	0.00	0.00	0.00	4.77	0.00	0.00	9.33	46.23			
	$TU_2^w$	0.04	0.05	0.06	5.23	0.20	0.06	6.96	75.53			
	$%G_2^w$	0.00	0.00	0.00	4.77	0.00	0.00	9.76	50.47			
	$CPU_2$	0.25	0.31	2.62	13.06	1.00	6.83	26.31	91.26			
	$#Opt_2$	4	4	4	3	4	4	2	0			
F3	$DL_3^w$	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00			
	$DU_3^w$	0.00	0.00	0.00 0.00 0.00		0.00	0.00	0.00	0.00			
	$%G_3^w$	0.00	.00 0.00 0.00 0.00		0.00	0.00	0.00	0.00	0.00			
	$CPU_3$	1.52	1.73	53.83	407.46	7.03	2357.58	650.40	576.25			
	#Opt <sub>3</sub>			4	4 4 4							

- Deviations of the upper bounds from optimal values:  $DU_i^w =$  $100(U_i^w - W^*)/W^*, i = 0, ..., 3.$ • Optimality gaps:  ${}^{\mathcal{K}}G_i^w = 100(U_i^w - L_i^w)/L_i^w, i = 0, ..., 3.$

Table 3 gives average values, over all the instances in each of the classes  $C_{n_4,c_s}$ , of the above percentage deviations and gaps, as well as the computing times (in seconds) required to obtain each of the corresponding bounds. A maximum computing time of 3600 seconds was established in all the tests, with the exception of instance 21\_70\_4B\_E\_S with formulation F3 for which, as mentioned, we allowed to exceed that time limit in order to guarantee the optimality of the obtained solution. The table also gives the total number of instances of the corresponding class solved to proven optimality in each case. The heading of the table consists of three rows indicating the number of calling ships (n), the parameter combinations (S/L-A/E), and the labels of the instances in the corresponding class (Inst. labels), respectively. As can be seen, each instance class  $\mathcal{C}_{n\_4\_c\_s}$  is associated with one column. The numerical results are summarized in four blocks of rows, the first one, labeled HEUR, for the heuristic solution combined with the LP bound of F1, is followed by one block for each of the formulations F1, F2 and F3, each of them labeled as  $F_i$ , i = 1, ..., 3, respectively. Blocks HEUR,  $F_1$  and  $F_3$  have the same structure consisting of five rows; the first three rows refer to  $DL_i^w$ ,  $DU_i^w$  and  $G_i^w$ , respectively, row CPU<sub>i</sub> to computing times, and row  $\#Opt_i$ gives the number of instances in each class optimally solved within the maximum computing time. Block F<sub>2</sub> consists of seven rows, the first two ones related to the outcome of formulation F2: average percentage deviations %DL<sub>2</sub><sup>w</sup>, and average computing times for optimally solving F2  $(TL_2)$ . The next two rows are related to the outcome of the second phase of the algorithmic procedure we have explained: average percentage deviations of the obtained upper bounds ( $\%DU_2^w$ ) and average computing times ( $TU_2$ ). The final three rows give the average percentage gaps at termination ( $%G_2^w$ ), the total computing times ( $CPU_2 = TL_2 + TU_2$ ), and the number of instances solved to proven optimality (#Opt<sub>2</sub>), which is given by the number of instances in each class for which the optimality

check based Proposition 2 indicated that the solution produced by F2 was feasible for SBTP.

At a first glance, the results of HEUR may seem quite modest, although a closer look highlights the following positive aspects: the simplicity of the procedures used to obtain the lower and upper bounds, the quality of the LP bounds produced by formulation F1, and the small computing times required to obtain these (initial) lower and upper bounds. Indeed these results are outperformed by those of F1, although it is somehow disappointing that the good quality of the initial LP bounds does not result in a more effective exploration of the enumeration tree. As can be seen, only 10 out of these 32 benchmark instances were optimally solved within the maximum time limit of one hour. Note that all the instances optimally solved belong to classes where the type of service parameter is s = S, i.e., they have *small* service times. Among the instances with large service times those corresponding with composition A (where the proportion of the different types of ships is not the same) produced somewhat tighter lower bounds at termination; in particular those lower bounds coincided with the optimal value for two additional instances in class  $\mathcal{C}_{50\_4\_A\_L}$  and one additional instance in  $\mathcal{C}_{70\_4\_A\_L}$ . Nevertheless, the overall results indicate that while F1 produces very tight lower bounds in small computing times, it has difficulties in producing feasible solutions of good quality. In fact, for several unsolved instances, the upper bound at termination was associated with a solution found by the default heuristic at the root node.

On the contrary, the results shown in the block F2 indicate the effectiveness of the 2-phase solution procedure based on formulation F2. On the one hand, F2 produces extremely tight lower bounds, which already correspond to SBTP optimal values for 15 and 10 out of the 16 instances with n = 50 and n = 70, respectively analyzed in Table 3. As can be seen, the value of  $%DL_2^w$  for the class  $C_{50_4LE}$  is 0, which means that for instance 11, which is the only instance of  $C_{50\_4\_L\_E}$  where the outcome of F2 was not feasible for SBTP, the obtained lower bound coincided with the optimal SBTP value. The values of  $%DL_2^w$  for the classes  $C_{70 4 L A}$  and  $C_{70 \ 4 \ L \ E}$  are a little higher: 0.39 and 2.82, respectively. For  $C_{70 \ 4 \ L \ A}$ there is again one single instance (the one labeled 31) for which

Summarv	of numerical	results with	F2 for	instances	with $b \in$	{4.8	12}	and $n \in$	{70.	100.	150}
Jummary	of numerical	icouito with	12 101	motanees	WITH D C	ן ד, ט,	121	and $n \in$	ι <i>ν</i> υ,	100,	150).

b	п	S/L	A/E	Inst.	$%DL_2^w$	TL <sub>2</sub>	$%DU_2^w$	TU <sub>2</sub>	$%G_2^w$	CPU <sub>2</sub>	#Opt <sub>2</sub>
4	100	S	А	33-36	0.00	12.14	3.75	2.01	3.75	14.15	3
			Е	37-40	0.00	16.97	0.00	0.08	0.00	17.05	4
		L	Α	45-48	0.29	25.48	31.46	8.77	31.91	34.24	1
			Е	41-44	2.03	20.72	34.01	1.94	36.91	22.66	0
8	70	S	А	61-64	0.00	0.30	0.00	0.12	0.00	0.41	4
			Е	57-60	0.00	0.32	0.00	0.12	0.00	0.45	4
		L	Α	49-52	0.00	0.43	0.00	0.15	0.00	0.57	4
			Е	53-56	0.00	0.46	0.00	0.19	0.00	0.65	4
	100	S	Α	65-68	0.00	0.42	0.00	0.16	0.00	0.58	4
			Е	77-80	0.00	0.52	0.00	0.12	0.00	0.64	4
		L	Α	69-72	0.00	1.43	0.00	0.26	0.00	1.70	4
			Е	73–76	0.07	10.67	2.84	225.56	2.92	236.22	1
12	150	S	А	93-96	0.00	0.77	0.00	0.21	0.00	0.98	4
			Е	81-84	0.00	0.83	0.00	0.33	0.00	1.16	4
		L	Α	89-92	0.00	1.65	0.00	4.18	0.00	5.83	4
			Е	85-88	0.00	6.69	0.00	14.55	0.00	21.23	4

Table 5

Summary of numerical results with *F*3 for instances with  $b \in \{4, 8, 12\}$  and  $n \in \{70, 100, 150\}$ .

b	п	S/L	A/E	Inst.	$%DL_3^w$	$%DU_3^w$	$%G_3^w$	CPU <sub>3</sub>	#Opt₃
4	100	S	А	33-36	0.00	0.00	0.00	848.24	4
			Е	37-40	0.00	0.00	0.00	1112.21	4
		L	Α	45-48	0.00	0.00	0.00	401.28	4
			Е	41-44	0.00	0.00	0.00	522.88	4
8	70	S	А	61-64	0.00	0.00	0.00	3.19	4
			E	57-60	0.00	0.00	0.00	3.51	4
		L	Α	49-52	0.00	0.00	0.00	5.99	4
			Е	53-56	0.00	0.00	0.00	6.90	4
	100	S	Α	65-68	0.00	0.00	0.00	5.75	4
			Е	77-80	0.00	0.00	0.00	6.64	4
		L	Α	69-72	0.00	0.00	0.00	1030.30	4
			Е	73–76	0.00	0.00	0.00	3258.72	4
12	150	S	А	93-96	0.00	0.00	0.00	9.45	4
			E	81-84	0.00	0.00	0.00	12.76	4
		L	Α	89-92	0.00	0.00	0.00	22.46	4
			Е	85-88	0.10	1.60	1.70	7173.10	2

the lower bound produced by F2 did not coincide with the optimal SBTP value. Instead, none of the lower bounds obtained for the instances of  $C_{70\_4\_L\_E}$  coincided with their optimal values, and their percentage deviations range in 2.04–4.63. We would like to call the attention on the computing times needed to optimally solve F2, which for all instances with n = 50 and n = 70 were smaller than 12 and 26 seconds, respectively.

Nevertheless, the best results were clearly obtained with formulation F3, which produced proven optimal solutions for all the considered instances. The computing times are remarkable. All instances in classes  $C_{50\_4\_S\_A}$  and  $C_{50\_4\_S\_E}$  were solved in less than 2.5 seconds; the computing times of instances in  $C_{50\_4\_L\_A}$  range in 2.9–73.1 seconds, except for instance 15, which required 128.4 seconds. Instances in  $C_{50\_4\_L\_E}$  were solved in less than 275 seconds, with the exception of instance 9, which required 128.4 seconds. As could be expected, computing times increase with the number of calling ships, although the computing times are still very small. Only two out of the 16 instances with n = 70 required more than 1000 seconds: instance 21, which, as mentioned, consumed 9,368.14 seconds, and instance 31, which consumed 1,091.5 seconds.

### 6.2. Numerical results for larger instances

Next we present the results we have obtained with the instances with  $b \in \{8, 12\}$ , all of which have a number of calling ships  $n \in \{70, 100, 150\}$ . Taking into account the results obtained

with the smaller instances, now we have tested the 2-phase algorithm based on F2 and formulation F3. While we did not set a maximum time limit for the 2-phase algorithm, as for all instances the procedure terminated in small computing times, we did set a maximum time limit of 10,800 seconds (three hours) for the solution of formulation F3. To facilitate the readability of the numerical results they are summarized in two different tables: Table 4 for the 2-phase procedure based on F2 and Table 5 for the results obtained with formulation F3. The structure of both tables is similar: each row corresponds to a class of instances and, except for the columns showing instance characteristics, there is one column for each of the items analyzed. Hence, Table 4 has seven such columns, respectively labeled with %DL2, TL2, %DU2, TU2, %G2, CPU2, and #Opt2, whereas Table 5 has five such columns, respectively labeled with  $DL_3^w$ ,  $DU_3^w$ ,  $G_3^w$ ,  $CPU_3$ , and  $HOpt_3$ . The meaning of the headings is the same as in Table 3.

We can again appreciate the excellent performance of both the 2-phase solution procedure and formulation F3. The algorithmic scheme based on F2 produced a provable optimal solution for 53 out of the 64 larger instances, and for the instances where an optimal solution was not found the percentage deviations  $\%DL_2^w$  are extremely small. The largest such deviations appear in  $C_{100,4,L,E}$ , in particular for instance 44\_50\_4B\_E\_L, where the percentage deviation of the lower bound produced by F2 and the optimal value is 3.85. Other classes of instances where optimal SBTP solutions were not always found are  $C_{100,4,S,A}$ ,  $C_{100,4,L,A}$ , and  $C_{100,8,L,E}$ . Still, for most instances where F2 did not produce an optimal solution,

Table 6					
Comparison	of	F2,	F3	and	GSP+

					F2			F3			$GSP^+$			
b	n	S/L	A/E	Inst.	%G <sub>2</sub> <sup>w</sup>	CPU	#Opt <sub>2</sub>	$%G_3^w$	CPU	#Opt <sub>3</sub>	$%G_I^w$	CPU	#np_Opt	#Opt
4	50	S	А	1-4	0.00	0.25	4	0.00	1.52	4	0.00	5.88	0	4
			Е	5-8	0.00	0.31	4	0.00	1.73	4	0.00	6.23	0	4
		L	А	13-16	0.00	2.62	4	0.00	53.83	4	9.92	8103.40	2	1
			Е	9-12	4.77	13.06	3	0.00	407.46	4	77.23	10,800.00	1	0
	70	S	Α	17-20	0.00	1.00	4	0.00	7.03 4		4.63	2709.45	0	3
			E	21-24	0.00	6.83	4	0.00	2357.58	4	0.00	673.98	0	4
		L	Α	29-32	9.76	26.31	2	0.00	650.40	4	9.82	4703.08	0	3
			Е	25-28	50.47	91.26	0	0.00	576.25	4	3.94	3330.65	1	3
	100	S	Α	33-36	3.75	14.15	3	0.00	848.24	4	12.06	7189.75	1	2
			Е	37-40	0.00	17.05	4	0.00	1112.21	4	0.00	2525.68	0	4
		L	Α	45-48	1.91	34.24	1	0.00	401.28	4	5.89	4381.63	1	3
			Е	41-44	6.91	22.66	0	0.00	522.88	4	8.33	3526.25	1	3
8	70	S	Α	61-64	0.00	0.41	4	0.00	3.19	4	0.00	15.43	0	4
			Е	57-60	0.00	0.45	4	0.00	3.51	4	0.00	17.40	0	4
		L	Α	49-52	0.00	0.57	4	0.00	5.99	4	0.00	23.45	0	4
			Е	53-56	0.00	0.65	4	0.00	6.90	4	0.00	24.95	0	4
	100	S	Α	65-68	0.00	0.58	4	0.00	5.75	4	0.00	23.95	0	4
			Е	77-80	0.00	0.64	4	0.00	6.64	4	0.00	26.35	0	4
		L	Α	69-72	0.00	1.70	4	0.00	1030	30.4	0.00	337.20	0	4
			Е	73-76	2.92	236.22	1	0.00	3258.72	4	951.74	10,800.00	0	0
12	150	S	Α	93-96	0.00	0.98	4	0.00	9.45	4	0.00	64.50	0	4
			Е	81-84	0.00	1.16	4	0.00	12.76	4	0.00	93.78	0	4
		L	Α	89-92	0.00	5.83	4	0.00	22.46	4	0.00	182.50	0	4
			Е	85-88	0.00	21.23	4	1.70	7173.10	2	25.21	7437.80	0	2
# Ins	stances	Optima	lly solv	ed in total:			78			94			7	76

the obtained lower bound coincides with the optimal SBTP value, the only exceptions being instance 46\_100\_4B\_E\_L with a percentage deviation of 1.18 and instance 76 with a percentage deviation of 0.29.

While the lower bounds produced by F2 are optimal or quasioptimal, the quality of the upper bounds is not so high for instances where the optimality check did not certify the optimality of the obtained solution. This is not surprising, given the simplicity of the second phase, which produces a feasible solution in which the assignment of served ships to berths is dictated by the outcome of  $AP(\bar{z}, \bar{h})$ , despite the fact that the optimality check has tested negative, which is a rather clear indication that such assignment is probably not an optimal one. Still, the upper bounds that we obtain in such cases are, in general, quite tight, with the exception of those for instances in classes  $C_{100_4LA}$  and  $C_{100_4LE}$ .

Note that all the instances that were not optimally solved with the procedure based on F2 correspond to classes with high values of the demand ratio R = D/B. In particular, for  $C_{100\_4\_L\_E}$ , which produced the largest percentage deviation gaps, R = 1.97 (the overall demand is nearly twice as service capacity), which is the largest value among all classes. Classes  $C_{100\_4\_S\_A}$ ,  $C_{100\_4\_L\_A}$ , and  $C_{100\_8\_L\_E}$  also have values of R > 1.

We finally observe that the total computing times required by the 2-Phase procedure are remarkably small. Average total computing times are always below 250 seconds, even for the most demanding class in that sense,  $C_{100_{-}8_{-}L_{-}E}$ . Notice that the average computing times of the first phase in which formulation F2 is solved to optimality, TL<sub>2</sub>, are below 26 seconds for all classes of instances. The most demanding individual instance for the first phase was instance 45, which required less than 35 seconds. The first phase consumed less than five seconds for all instances with n = 150, with the exception of instance 85, which required nearly 16 seconds. In fact the computing load of the 2-phase procedure relies essentially on the second phase and, in particular, on the solution on the berth allocation problem  $AP(\bar{z}, \bar{h})$ , which becomes more demanding, not only as the sizes of the instances increase, but mainly as the demand ratio R increases. We can observe that the largest average computing time for the second phase

of 236.22 seconds corresponds again to class  $C_{100\_8\_L\_E}$ , which, as mentioned, has the largest value of *R*.

We now focus our attention on the results of formulation F3, which are summarized in Table 5. As can be seen, 62 out of the 64 instances of the considered classes were solved to proven optimality within the maximum time limit of 10,800 seconds. The only two instances that were not solved to optimality belong to class  $C_{150_{-12}_{-L_{-E}}}$ , namely instances 85 and 86. Since both of these instances were optimally solved with the 2-phase procedure, we know their optimal values, so the obtained results can be better assessed. In particular, their optimal total waiting times are 440 and 362, respectively. At termination, the lower and upper bounds on the total waiting time that we obtained for instance 85 are  $L_3^w = 439.30$  and  $U_3^w = 467$ , respectively, with corresponding percentage deviations of  $DL_3^w = 0.16\%$  and  $DU_3^w = 6.14\%$ . For instance 86 the obtained bounds are  $L_3^w = 361.16$  and  $U_3^w = 363$ , resulting in percentage deviations of  $DL_3^w = 0.23\%$  and  $DU_3^w = 0.51\%$ . Thus, in both cases by rounding up the lower bound we obtain the optimal values. While the upper bound of instance 86 differs in just one unit from the optimal value, the best solution found for instance 85 has a value of 467, with a difference of 27 from the optimal value.

In general, the computing times needed to solve F3 are notably below the maximum time limit. Apart from instances 85– 86, which reached the limit, only three instances (71, 75 and 76) required more than one hour of computing time; their respective computing times being 4044, 4376 and 6043 seconds.

Similarly to what we have observed with the 2-phase algorithm, the difficulty for solving an instance clearly depends on its demand rate: the higher average computing times are, in general again associated with instances with values of *R* very close to 1. The two instances that reached the limit belong have n = 150, b = 12, i.e. 228,000 binary variables  $\hat{h}$ , and have a demand rate R = 0.96, which, is quite close to 1, and is the largest demand rate among all classes of instances with n = 150. This can be clearly appreciated in Fig. 3 where we have plotted the computing times of the full set of individual instances both for F2 and F3.

#### E. Fernández and M. Munoz-Marquez



Fig. 3. Computing times for F2, F3 for the full set of benchmark instances.



**Fig. 4.** Average computing times for F2, F3 and  $GSP^+$ , for instances with b = 4.

A detailed look at the individual lower and upper bounds for each of the instances (see Tables 7–12 in the Appendix) highlight the remarkable quality of the lower bounds obtained with both F2 and F3, which already are the optimal value or deviate very few units from it. Moreover, these bounds (or others with less than one unit of difference from them) are usually attained already at the root node of the enumeration tree. Obtaining optimal or nearoptimal feasible solutions is usually more demanding although the obtained results are equally satisfactory, particularly those of F3, which solved to proven optimality all but two instances. In total only five out of the 96 considered instances consumed more than one hour.

### 6.3. Comparison of F2 and F3 with the results of Iris et al. (2018)

We conclude this section with a comparison of our numerical results with those of Iris et al. (2018) with the same set of benchmark instances. In particular, we compare F2 and F3 with the so-called formulation  $GSP^+$ , which produced the best results among the alternatives tested computationally in Iris et al. (2018). The comparison is summarized in Table 6, where the full set of instances is considered and each row corresponds to a class of instances. The table contains three blocks of columns, for the 2phase algorithm based on F2, for formulation F3, and for formulation  $GSP^+$ , respectively. Each block has a first column for average percentage optimality gaps at termination (labeled  $%G^w$ ), a second column for average computing times (labeled *CPU*), and a final column showing the number of instances solved to proven optimality (#Opt). The block *GSP*<sup>+</sup> contains another column (#np\_Opt), just before the final one, indicating the number of instances for which the best solution found was optimal, although its optimality could not be proven within the allowed computing time. Since the deviations reported in Iris et al. (2018) are computed relative to the overall objective function value, from the results reported in the paper, for each instance we have computed the deviations  $%G^w$  relative to the waiting times, by subtracting from the reported objective function values the constant penalty value *P*<sup>\*</sup>.

The results of Table 6 show that both the 2-phase procedure based on F2 and F3 outperform  $GSP^+$  in terms of the number of solutions found whose optimality could be proven. This superiority is particularly relevant in terms of the computational effort required to obtain the results, as can be observed in Figs. 4 and 5, where instances have been grouped by classes with the same parameter values and the horizontal axis indicates the average demand ratio R = D/C. In all the figures the vertical axis considers a maximum of 18,000 seconds, except for the comparison for n = 70 and b = 8 (left most image of Fig. 5) where the maximum of the vertical axis is only 60 seconds, because the computing times of all instances and compared formulations were always below that time limit.



**Fig. 5.** Average computing times for F2, F3 and  $GSP^+$ , for instances with b = 8, 12.

### 7. Conclusions

In this paper we have studied the SBTP in which there is a set of ships cyclically calling for service at a port. The STBP combines strategic decisions for selecting the ships to serve and operational decisions for setting the service times for the selected ships, with the objective of minimizing a penalty for the rejected ships plus the total sum of the waiting times of the accepted ships. Several formulations have been developed. All of them use binary variables that classify served ships depending on whether or not their service starts during their arrival cycle or in the next one. This helps modeling the STBP, since a closed linear expression can be obtained for the waiting times. The most basic formulations presented, which, in addition use predecessor variables to identify the sequence of calling ships served consecutively in the same berth, are outperformed by alternative formulations that avoid such variables, in which alternative binary decision variables are defined to determine the actual time period in which service to each ship starts. Two such alternatives have been introduced and studied: F2 where the new decision variables are aggregated over all berths, and F3 where variables consider in addition the index of the berth where the ship is served. While F2 is a relaxation of SBTP, it can be solved in remarkably small computing times and, together with a simple check that indicates whether or not its optimal solution is also optimal for SBTP, can be used very effectively to produce optimal or near-optimal SBTP solutions in a 2-phase solution algorithm. F3 is an exact formulation that produces SBTP solutions of guaranteed optimality. The proposed formulations have been computationally tested on a set of benchmark instances from the literature. The obtained numerical results assess the efficiency of the 2-phase solution procedure based on F2 and on formulation F3. Both alternatives outperform the so-called formulation GSP<sup>+</sup> of Iris et al. (2018), which is the best SBTP formulation in the literature, both in terms of the number of provable optimal solutions that they produce and the computing time requirements.

The proposed formulations can be extended in several ways to deal with more general versions of the SBTP. At the strategic level, one can easily incorporate requirements imposing that a given subset of ships must be necessarily served, either by fixing at value one their associated  $z_k$  variables or by increasing arbitrarily their rejection penalty. Another straightforward extension is to consider ship-dependent penalties for the rejected ships, not necessarily proportional to their service times.

A more challenging extension is to consider situations in which the number of available berths is time dependent. This would allow to consider periods of lower activity like, for instance, Sundays. Let  $b_t$ ,  $t \in T$  denote the number of berths available at time period t, then the right hand side of the berths availability constraint (9a) at time period  $t \in T$  of formulation F2 can be easily substituted by  $b_t$ . Formulation F3 can also be easily adapted by considering a timedependent index set of available berths  $R_t = \{1, ..., b_t\}$ ,  $t \in T$ , and by stating the feasibility constraints (16a), for  $r \in R_t$ ,  $t \in T$ .

Finally, a further non-trivial extension of the SBTP would be to consider scenarios where different calling companies have different cycle lengths. In its turn this would require to propose new formulations as the current ones are no longer valid.

### Acknowledgments

This research was partially funded by the Spanish Ministry of Economy and Competitiveness and ERDF funds [Grant PID2019-105824GB-100 (MINECO/FEDER)]. This support is gratefully ac-knowledged. The authors are thankful to Eduardo Lalla-Ruiz who kindly made available to us the set of benchmark instances.

### Appendix A. Tables with detailed results

In this appendix we give detailed results for the individual instances obtained with each of the tested formulations and solution alternatives, as well as for the formulation  $GSP^+$  of Iris et al. (2018). In all cases, the bounds that we analyze correspond to waiting times. That is, if the value of a solution is  $W + P^*$ , where  $W = \sum_{i \in \overline{V}} w_i$  is the total waiting time of the served ships and  $P^*$ the constant penalty term, we ignore the term  $P^*$ , and only consider the value W. Otherwise, relatively large differences in the value of W may somehow be hidden behind the large value of the penalty term  $P^*$ . In all cases, the bounds on the overall objective function value can be obtained by adding the penalty value  $P^*$ .

Tables 7 and 8 refer to instances with four berths and a number of calling ships n = 50 and n = 70, respectively. Both tables have a similar structure. The meaning of the columns is as follows: Entries in bold indicate best-known results.

- The first three columns give the numerical instance label, the value of the penalty  $(P^*)$ , and the optimal value of the total waiting time  $(W^*)$  of the instance, respectively.
- Block *HEUR* refers to the feasible solution obtained with the procedure described in Section 5.5 in which a variation of  $AP(\bar{z}, \bar{h})$  is solved with  $\overline{V} = V$ ,  $\bar{h}_{it} = 1$  if an only if  $t = a_i$ ,  $i \in V$ , Constraints (15b) stated as " $\leq$ " inequalities, and the additional mother-ship constraints  $\sum_{r \in R} \lambda_{ir} = z_{k(i)}$ ,  $i \in V$ . In the objective function, the weight that has been used for the combination of the *service* and *overlap* criteria is  $\mu = 100$ .
- The values reported in this block are  $U_0^w$ ,  $\&U_0^w = 100(U_0^w W^*)/W^*$ , and  $T_0$ , for the overall waiting time, its percentage deviation from the optimal total waiting time, and the computing time required to obtain the solution respectively.
- Blocks *F*1, *F*2, *F*3 and *GSP*<sup>+</sup> for the results of formulations F1, F3 and *GSP*<sup>+</sup> formulation of Iris et al. (2018), respectively. The values reported in these blocks are in each case  $L_i^w$ ,  $U_i^w$ ,  $%G_0^w = 100(U_i^w - L_i^w)/L_i^*$ , and  $T_i$ , for the overall waiting times of

Results on instances with 50 ships and 4 berths.

	<b>P</b> *	w*	HEU	R		F1	F1 F2						F3				GSP <sup>+</sup>								
			$\overline{U_0^w}$	%U_0 <sup>w</sup>	T <sub>0</sub>	$L_1^w$	$U_1^w$	$%G_1^w$	<i>T</i> <sub>1</sub>	$L_2^w$	$%L_2^w$	$T_{L_2}$	Feas.	$U_2^w$	$%U_{2}^{w}$	$T_{U_2}$	<i>T</i> <sub>2</sub>	$L_3^w$	$U_3^w$	$%G_3^w$	T <sub>3</sub>	$\overline{L_I^w}$	$U^w_I$	$%G_I^w$	$T_I$
1	0	11	11	0.00	0.5	11.0	11	0.00	1.0	11	0.00	0.2	Y	11	0.00	0.04	0.2	11.0	11	0.00	1.4	11.0	11	0.00	5.6
2	0	13	16	23.00	0.6	13.0	13	0.00	3.2	13	0.00	0.2	Y	13	0.00	0.04	0.2	13.0	13	0.00	2.0	13.0	13	0.00	6.2
3	0	16	23	43.75	0.5	16.0	16	0.00	3.0	16	0.00	0.2	Y	16	0.00	0.05	0.3	16.0	16	0.00	1.4	16.0	16	0.00	6.1
4	0	14	18	28.57	0.5	14.0	14	0.00	3.0	14	0.00	0.2	Y	14	0.00	0.04	0.3	14.0	14	0.00	1.3	14.0	14	0.00	5.6
5	0	22	25	13.64	0.4	22.0	22	0.00	1.5	22	0.00	0.2	Y	22	0.00	0.05	0.3	22.0	22	0.00	1.4	22.0	22	0.00	6.1
6	0	23	38	65.22	0.6	23.0	23	0.00	4.5	23	0.00	0.3	Y	23	0.00	0.04	0.3	23.0	23	0.00	1.7	23.0	23	0.00	6.3
7	0	62	97	56.45	1.2	62.0	62	0.00	17.6	62	0.00	0.3	Y	62	0.00	0.04	0.3	62.0	62	0.00	2.2	62.0	62	0.00	6.4
8	0	23	28	21.74	0.4	23.0	23	0.00	3.3	23	0.00	0.3	Y	23	0.00	0.05	0.3	23.0	23	0.00	1.6	23.0	23	0.00	6.1
9	300,000	166	372	124.10	7.2	156.6	372	137.59	3599.7	166	0.00	11.6	Y	166	0.00	0.04	11.7	166.0	166	0.00	945.2	149.9	172	13.31	10800.0
10	270,000	236	494	109.32	9.0	236.0	494	109.32	3599.4	236	0.00	8.7	Y	236	0.00	0.04	8.7	236.0	236	0.00	374.9	211.0	236	10.59	10800.0
11	350,000	728	914	25.55	27.4	726.1	914	25.89	3600.2	728	0.00	8.7	Ν	867	19.09	20.79	29.4	728.0	728	0.00	250.0	359.9	1012	89.57	10800.0
12	140,000	252	371	47.22	182.7	252.0	371	47.22	3600.0	252	0.00	2.4	Y	252	0.00	0.05	2.5	252.0	252	0.00	59.7	142.2	286	57.06	10800.0
13	0	291	440	51.20	6.8	291.0	423	45.36	3600.1	291	0.00	6.6	Y	291	0.00	0.05	6.6	291.0	291	0.00	73.1	259.6	291	10.79	10800.0
14	0	228	319	39.91	2.9	228.0	313	37.28	3599.5	228	0.00	1.3	Y	228	0.00	0.07	1.3	228.0	228	0.00	10.9	213.9	231	7.50	10800.0
15	0	216	351	62.50	6.3	215.1	351	63.24	3599.4	216	0.00	2.1	Y	216	0.00	0.06	2.2	216.0	216	0.00	128.4	180.6	216	16.39	10800.0
16	0	107	167	56.07	1.9	107.0	136	27.10	3602.1	107	0.00	0.3	Y	107	0.00	0.06	0.4	107.0	107	0.00	2.9	107.0	107	0.00	13.6

#### Table 8

Results on instances with 70 ships and 4 berths.

ŀ	<b>)</b> *	<i>w</i> *	HEU	R		F1				F2								F3				GSP+			
			$\overline{U_0^w}$	%U <sub>0</sub> <sup>w</sup>	T <sub>0</sub>	$L_1^w$	$U_1^w$	$%G_1^w$	<i>T</i> <sub>1</sub>	$L_2^w$	$%L_2^w$	$T_{L_2}$	Feas.	$U_2^w$	%U <sub>2</sub> <sup>w</sup>	$T_{U_2}$	<i>T</i> <sub>2</sub>	$L_3^w$	$U_3^w$	$%G_3^w$	T <sub>3</sub>	$\overline{L_I^w}$	$U^w_I$	%G <sup>w</sup>	T <sub>I</sub>
17	0	239	349	46.03	12.9	239.0	331	38.49	3599.6	239	0.00	1.4	Y	239	0.00	0.07	1.47	239.0	239	0.00	14.9	205.0	243	15.90	10800.0
18	0	56	82	46.43	0.8	56.0	61	8.93	3599.7	56	0.00	0.6	Y	56	0.00	0.08	0.66	56.0	56	0.00	3.5	56.0	56	0.00	13.1
19	0	64	85	32.81	0.7	64.0	64	0.00	70.9	64	0.00	0.5	Y	64	0.00	0.60	1.10	64.0	64	0.00	2.3	64.0	64	0.00	8.6
20	0	104	146	40.38	0.6	104.0	104	0.00	114.9	104	0.00	0.7	Y	104	0.00	0.07	0.76	104.0	104	0.00	7.4	104.0	104	0.00	16.1
21	0	612	739	20.75	2.9	608.5	734	20.63	3599.7	612	0.00	22.7	Y	612	0.00	0.07	22.74	612.0	612	0.00	9368.1	612.0	612	0.00	2384.1
22	0	217	313	44.24	1.2	217.0	293	35.02	3599.7	217	0.00	0.9	Y	217	0.00	0.06	1.02	217.0	217	0.00	6.4	217.0	217	0.00	22.1
23	0	331	407	22.96	3.8	331.0	404	22.05	3600.2	331	0.00	1.0	Y	331	0.00	0.06	1.03	331.0	331	0.00	19.6	331.0	331	0.00	24.5
24	0	215	343	59.53	8.1	214.8	340	58.27	3599.8	215	0.00	2.4	Y	215	0.00	0.06	2.52	215.0	215	0.00	36.2	215.0	215	0.00	265.2
25 2	2,500,000	147	362	146.26	6.9	142.6	362	153.79	3599.3	144	2.04	24.2	N	206	40.14	8.84	33.03	147.0	147	0.00	768.3	127.0	147	13.61	10800.0
26 2	2,420,000	86	196	127.91	3.9	82.2	196	138.50	3600.3	84	2.33	9.4	N	130	51.16	227.31	236.69	86.0	86	0.00	534.9	86.0	86	0.00	778.9
27 2	2,590,000	87	167	91.95	3.4	80.6	167	107.22	3599.7	85	2.30	25.5	N	133	52.87	9.80	35.25	87.0	87	0.00	609.3	87.0	87	0.00	693.5
28 2	2,740,000	108	258	138.89	6.7	103.0	258	150.56	3600.1	103	4.63	3.9	N	152	40.74	56.16	60.07	108.0	108	0.00	392.5	108.0	108	0.00	1050.2
29 1	,410,000	153	305	99.35	3.5	150.1	305	103.25	3602.1	153	0.00	23.5	Y	153	0.00	23.55	47.05	153.0	153	0.00	423.7	112.0	156	28.76	10800.0
30 1	,480,000	100	230	130.00	2.9	90.5	230	154.06	3600.2	100	0.00	24.3	N	128	28.00	2.30	26.56	100.0	100	0.00	583.3	100.0	100	0.00	6436.3
31 1	,290,000	129	301	133.33	2.3	123.0	301	144.68	3599.8	127	1.55	11.4	Ν	141	9.30	1.90	13.29	129.0	129	0.00	1091.5	129.0	129	0.00	658.1
32 1	,290,000	183	439	139.89	5.1	179.9	439	144.08	3599.9	183	0.00	18.3	Y	183	0.00	0.07	18.35	183.0	183	0.00	503.1	183.0	183	0.00	917.9

#### Table 9

Results on instances with 100 ships and 4 berths.

	P*	<i>w</i> *	F2	F2							F3				GSP+				
			$L_2^w$	$%L_2^w$	$T_{L_2}$	Feas.	$U_2^w$	$%U_2^w$	$T_{U_2}$	<i>T</i> <sub>2</sub>	$L_3^w$	$U_3^w$	$%G_3^w$	<i>T</i> <sub>3</sub>	$L_I^w$	$U_I^w$	$%G_I^w$	T <sub>I</sub>	
33	770,000	219	219	0.00	16.59	Y	219	0.00	0.07	16.6	219.0	219	0.00	429.9	219.0	219	0.00	3776.1	
34	620,000	132	132	0.00	13.19	Y	132	0.00	0.07	13.3	132.0	132	0.00	560.6	127.0	132	3.94	10800.0	
35	640,000	180	180	0.00	7.32	Y	180	0.00	0.09	7.4	180.0	180	0.00	719.2	180.0	180	0.00	3382.9	
36	700,000	300	300	0.00	11.48	Ν	345	15.00	7.80	19.3	300.0	300	0.00	1683.3	224.5	324	44.32	10800.0	
37	2,240,000	86	86	0.00	22.95	Y	86	0.00	0.08	23.0	86.0	86	0.00	766.0	86.0	86	0.00	773.1	
38	1,880,000	88	88	0.00	6.81	Y	88	0.00	0.09	7.0	88.0	88	0.00	887.3	88.0	88	0.00	2718.2	
39	1,840,000	115	115	0.00	25.71	Y	115	0.00	0.08	25.8	115.0	115	0.00	1289.2	115.0	115	0.00	3346.8	
40	2,140,000	145	145	0.00	12.42	Y	145	0.00	0.08	12.5	145.0	145	0.00	1506.4	145.0	145	0.00	3264.6	
41	5,760,000	42	41	2.38	19.34	Ν	53	26.19	1.21	20.5	42.0	42	0.00	272.3	42.0	42	0.00	842.6	
42	5,980,000	53	52	1.89	25.14	Ν	78	47.17	2.19	27.3	53.0	53	0.00	586.1	53.0	53	0.00	1029.9	
43	5,780,000	54	54	0.00	20.47	Ν	65	20.37	1.42	21.9	54.0	54	0.00	782.8	54.0	54	0.00	1432.5	
44	5,970,000	52	50	3.85	17.93	Ν	74	42.31	2.92	20.8	52.0	52	0.00	450.3	39.0	52	33.33	10800.0	
45	4,670,000	64	64	0.00	34.90	Ν	76	18.75	0.99	35.9	64.0	64	0.00	311.6	51.8	64	23.55	10800.0	
46	4,440,000	85	84	1.18	16.29	Ν	127	49.41	32.11	48.40	85.0	85	0.00	618.8	85.0	85	0.00	4010.6	
47	4,740,000	62	62	0.00	27.73	Y	62	0.00	0.09	27.82	62.0	62	0.00	375.9	62.0	62	0.00	924.3	
48	4,530,000	52	52	0.00	22.98	Ν	82	57.69	1.88	24.9	52.0	52	0.00	298.8	52.0	52	0.00	1791.6	

the lower and upper bounds obtained with the corresponding formulation, their percentage optimality gaps, and the computing times required to obtain the solution respectively. A time limit of one hour (3600 seconds) was set for F1, whereas this limit was of three hours for F3 and  $GSP^+$ .

In the block *GSP*<sup>+</sup> the bounds and deviations for the waiting times have been calculated from the results reported in Iris

et al. (2018). The computing times have been reproduced from the referenced paper.

• Block *F2* refers to the results of the 2-phase algorithm based on F2. Its first three columns give information on the output of the first phase:  $L_2^w$ ,  $\&l_2^w = 100(W^* - L_2^w)/W^*$ , and  $T_{L_2}$ , for the overall waiting time of the optimal solution of F2, its percentage deviation from the optimal waiting time, and the comput-

Results on instances with 70 ships and 8 berths.

	<b>P</b> *	w*	F2								F3				GSP <sup>+</sup>				
			$L_2^w$	$%L_2^w$	$T_{L_2}$	Feas.	$U_2^w$	$%U_2^w$	$T_{U_2}$	<i>T</i> <sub>2</sub>	$L_3^w$	$U_3^w$	$%G_3^w$	T <sub>3</sub>	$L_I^w$	$U_I^w$	$%G^w_I$	TI	
49	0	39	39	0.00	0.4	Y	39	0.00	0.13	0.6	39.0	39	0.00	5.8	39.00	39	0.00	26.4	
50	0	10	10	0.00	0.4	Y	10	0.00	0.17	0.6	10.0	10	0.00	5.3	10.00	10	0.00	20.1	
51	0	8	8	0.00	0.5	Y	8	0.00	0.14	0.6	8.0	8	0.00	5.8	8.00	8	0.00	19.9	
52	0	26	26	0.00	0.4	Y	26	0.00	0.15	0.6	26.0	26	0.00	7.0	26.00	26	0.00	27.4	
53	0	32	32	0.00	0.4	Y	32	0.00	0.20	0.6	32.0	32	0.00	7.1	32.00	32	0.00	28.1	
54	0	11	11	0.00	0.5	Y	11	0.00	0.20	0.7	11.0	11	0.00	6.8	11.00	11	0.00	21.1	
55	0	22	22	0.00	0.4	Y	22	0.00	0.20	0.6	22.0	22	0.00	6.6	22.00	22	0.00	28.6	
56	0	44	44	0.00	0.5	Y	44	0.00	0.17	0.7	44.0	44	0.00	7.2	44.00	44	0.00	22.0	
57	0	0	0	0.00	0.3	Y	0	0.00	0.13	0.5	0.0	0	0.00	3.4	0.00	0	0.00	16.0	
58	0	1	1	0.00	0.3	Y	1	0.00	0.14	0.5	1.0	1	0.00	3.4	1.00	1	0.00	20.1	
59	0	0	0	0.00	0.3	Y	0	0.00	0.12	0.4	0.0	0	0.00	3.7	0.00	0	0.00	18.1	
60	0	0	0	0.00	0.3	Y	0	0.00	0.11	0.4	0.0	0	0.00	3.6	0.00	0	0.00	15.4	
61	0	0	0	0.00	0.3	Y	0	0.00	0.12	0.4	0.0	0	0.00	3.3	0.00	0	0.00	16.2	
62	0	0	0	0.00	0.3	Y	0	0.00	0.13	0.4	0.0	0	0.00	3.2	0.00	0	0.00	15.2	
63	0	0	0	0.00	0.3	Y	0	0.00	0.11	0.4	0.0	0	0.00	3.1	0.00	0	0.00	14.3	
64	0	1	1	0.00	0.3	Y	1	0.00	0.11	0.4	1.0	1	0.00	3.2	1.00	1	0.00	16.0	

Tab	le	11

Results on instances with 100 ships and 8 berths.

	P*	<i>w</i> *	F2								F3				$GSP^+$			
			$L_2^w$	$%L_2^w$	$T_{L_2}$	Feas.	$U_2^w$	%U <sub>2</sub> <sup>w</sup>	$T_{U_2}$	<i>T</i> <sub>2</sub>	$L_3^w$	$U_3^w$	$%G_3^w$	<i>T</i> <sub>3</sub>	$L_I^w$	$U_I^w$	$G_{I}^{w}$	T <sub>I</sub>
65	0	0	0	0.00	0.4	Y	0	0.00	0.1	0.6	0.0	0	0.00	5.3	0.0	0	0.00	23.0
66	0	22	22	0.00	0.4	Y	22	0.00	0.2	0.6	22.0	22	0.00	6.1	22.0	22	0.00	25.8
67	0	4	4	0.00	0.4	Y	4	0.00	0.2	0.6	4.0	4	0.00	5.9	4.0	4	0.00	23.6
68	0	1	1	0.00	0.4	Y	1	0.00	0.2	0.6	1.0	1	0.00	5.7	1.0	1	0.00	23.4
69	0	74	74	0.00	0.6	Y	74	0.00	0.2	0.9	74.0	74	0.00	8.2	74.0	74	0.00	29.2
70	0	86	86	0.00	0.8	Y	86	0.00	0.3	1.1	86.0	86	0.00	9.2	86.0	86	0.00	54.5
71	0	304	304	0.00	2.2	Y	304	0.00	0.3	2.5	304.0	304	0.00	4044.3	304.0	304	0.00	1172.3
72	0	265	265	0.00	2.1	Y	265	0.00	0.3	2.4	265.0	265	0.00	59.4	265.0	265	0.00	92.8
73	160,000	469	469	0.00	9.6	Y	469	0.00	16.5	26.1	469.0	469	0.00	389.5	317.0	662	108.83	10800.0
74	490,000	267	267	0.00	13.7	Ν	279	4.49	362.2	375.9	267.0	267	0.00	2226.8	239.0	465	94.56	10800.0
75	80,000	718	718	0.00	10.1	Ν	742	3.34	71.9	82.0	718.0	718	0.00	4375.6	631.0	743	17.75	10800.0
76	80,000	339	338	0.29	9.2	Ν	351	3.54	451.6	460.8	339.0	339	0.00	6043.0	303.0	11,168	3585.81	10800.0
77	0	5	5	0.00	0.5	Y	5	0.00	0.1	0.7	5.0	5	0.00	6.5	5.0	5	0.00	25.6
78	0	7	7	0.00	0.5	Y	7	0.00	0.1	0.6	7.0	7	0.00	6.3	7.0	7	0.00	27.8
79	0	16	16	0.00	0.5	Y	16	0.00	0.1	0.6	16.0	16	0.00	6.5	16.0	16	0.00	27.6
80	0	1	1	0.00	0.5	Y	1	0.00	0.1	0.6	1.0	1	0.00	7.3	1.0	1	0.00	24.4

#### Table 12

Results on instances with 150 ships and 12 berths.

	<b>P</b> *	<i>w</i> *	F2								F3				GSP <sup>+</sup>				
			$L_2^w$	$%L_2^w$	$T_{L_2}$	Feas.	$U_2^w$	$%U_2^w$	$T_{U_2}$	<i>T</i> <sub>2</sub>	$L_3^w$	$U_3^w$	$%G_3^w$	<i>T</i> <sub>3</sub>	$L_I^w$	$U_I^w$	$G_{I}^{w}$	T <sub>I</sub>	
81	0	0	0.0	0.00	0.9	Y	0	0.00	0.3	1.1	0.0	0	0.00	11.1	0.0	0	0.00	98.7	
82	0	6	6.0	0.00	0.8	Y	6	0.00	0.3	1.1	6.0	6	0.00	13.2	6.0	6	0.00	86.2	
83	0	15	15.0	0.00	0.8	Y	15	0.00	0.3	1.1	15.0	15	0.00	13.8	15.0	15	0.00	89.9	
84	0	20	20.0	0.00	0.8	Y	20	0.00	0.5	1.3	20.0	20	0.00	12.9	20.0	20	0.00	100.3	
85	0	440	440.0	0.00	15.3	Y	440	0.00	7.5	22.7	439.3	467	6.31	10801.8	384.0	464	20.83	10800.0	
86	0	362	362.0	0.00	4.7	Y	362	0.00	12.1	16.8	361.2	363	0.51	10803.4	362.0	362	0.00	3511.0	
87	0	285	285.0	0.00	2.3	Y	285	0.00	17.5	19.8	285.0	285	0.00	782.9	285.0	285	0.00	4640.2	
88	0	250	250.0	0.00	4.4	Y	250	0.00	21.1	25.5	250.0	250	0.00	6304.3	165.0	297	80.00	10800.0	
89	0	74	74.0	0.00	1.1	Y	74	0.00	3.2	4.3	74.0	74	0.00	15.1	74.0	74	0.00	150.0	
90	0	87	87.0	0.00	2.4	Y	87	0.00	5.1	7.5	87.0	87	0.00	18.4	87.0	87	0.00	273.0	
91	0	22	22.0	0.00	1.0	Y	22	0.00	4.3	5.3	22.0	22	0.00	15.9	22.0	22	0.00	87.0	
92	0	312	312.0	0.00	2.1	Y	312	0.00	4.1	6.2	312.0	312	0.00	40.5	312.0	312	0.00	220.0	
93	0	0	0.0	0.00	0.8	Y	0	0.00	0.2	1.0	0.0	0	0.00	9.8	0.0	0	0.00	66.0	
94	0	0	0.0	0.00	0.8	Y	0	0.00	0.2	1.0	0.0	0	0.00	8.2	0.0	0	0.00	62.2	
95	0	0	0.0	0.00	0.8	Y	0	0.00	0.2	1.0	0.0	0	0.00	10.1	0.0	0	0.00	64.9	
96	0	1	1.0	0.00	0.7	Y	1	0.00	0.2	0.9	1.0	1	0.00	9.6	1.0	1	0.00	64.9	

ing times required to optimally solve F2 respectively. The next four columns refer to the second phase of the algorithm. The entries of *Feas.* are *Y* or *N* depending on whether or not the outcome of the feasibility check based on the auxiliary subproblem  $AP(\bar{z}, \bar{h})$  establishes the optimality for the SBTP of the solution of F2; columns  $U_2^w$ ,  $%U_2^w = 100(U_2^w - W^*)/W^*$ , and  $T_{U_2}$ , give the total waiting time of the feasible solution obtained in the sec-

ond phase of the algorithm, its percentage deviation from the optimal value  $W^*$ , and the computing time consumed in the second phase of the algorithm. The final column  $T_2 = T_{L_2} + T_{U_2}$  gives the total time.

Tables 9 –12 refer to the larger instances with b = 4 and n = 100 (Table 9), b = 8 and  $n \in \{70, 100\}$  (Tables 10 and 11), and b = 12 and n = 150 (Table 12). For these instances we report results

referring to formulations F2, and F3 with a maximum computing time of three hours. The meaning of the different columns is the same as explained above.

### 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(3), 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(3), 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(4), 461–473.
- Cordeau, J.-F., Laporte, G., Legato, P., & Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem. *Transportation Science*, 39(4), 526– 538. https://doi.org/10.1287/trsc.1050.0120.
- Du, J., & Leung, J. Y.-T. (1990). Minimizing total tardiness on one machine is np-hard. Mathematics of Operations Research, 15, 483–495.
- Garey, M. R., & Johnson, D. S. (1979). Computers and intractability: A guide to the theory of NP-completeness. Series of books in the mathematical sciences. W. H. Freeman.
- Hendriks, M., Laumanns, M., Lefeber, E., & Udding, J. (2010). Robust cyclic berth planning of container vessels. OR Spectrum, 32, 501–517.
- Huang, K., Suprayogi, & Ariantini (2016). A continuous berth template design model with multiple wharfs. *Maritime Policy & Management*, 43(6), 763–775.
- Imai, A., Nishimura, E., & Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4), 401–417.
- Imai, A., Nishimura, E., & Papadimitriou, S. (2003). Berth allocation with service priority. Transportation Research Part B: Methodological, 37(5), 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, Ç., Lalla-Ruiz, E., Lam, J., & Voss, S. (2018). Mathematical programming formulations for the strategic berth template problem. *Computers & Industrial Engineering*, 124, 167–179.
- Jin, J., 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.
- Korte, B., & Vygen, J. (2006). Combinatorial optimization. Springer-Verlag Berlin Heidelberg.
- delberg. Lalla-Ruiz, E., & Voss, S. (2016). Popmusic as a matheuristic for the berth allocation problem. *Annals of Mathematics and Artificial Intelligence*, 76, 173–189.
- Martello, S., & Toth, P. (1990). Knapsack problems; algorithms and computer implementations. Wiley-Interscience series in discrete mathematics and optimization. John Wiley and Sons.
- Monaco, M., & Sammarra, M. (2007). The berth allocation problem: A strong formulation solved by a Lagrangean approach. *Transportation Science*, 41(2), 265–280.
- Moorthy, R., & Teo, C.-P. (2006). Berth management in container terminal: The template design problem. *OR Spectrum*, *28*, 495–518.
- Wang, Q. M. Š., & Lee, C.-Y. (2016). Liner container assignment model with transittime-sensitive container shipment demand and its applications. *Transportation Research Part B: Methodological*, 90, 135–155.
- Wang, Z. L. S., & Qu, X. (2017). Weekly container delivery patterns in liner shipping planning models. Maritime Policy & Management, 44, 442–457.
- Xpress 2020. Fico® xpress solver. https://www.fico.com/es/products/ fico-xpress-solver.
- Xu, D., Li, C.-L., & Leung, J.-T. (2012). Berth allocation with time-dependent physical limitations on vessels. European Journal of Operational Research, 216(1), 47–56.
- Zhen, L, & Chang, D.-F. (2012). A bi-objective model for robust berth allocation scheduling. Computers & Industrial Engineering, 63, 262–273.