Models for the Discrete Berth Allocation Problem: A Computational Comparison

Katja Buhrkal[a], Sara Zuglian[b], Stefan Ropke[b], Jesper Larsen[c], Richard Lusby[c]

[a]*Department of Informatics and Mathematical Modelling, Technical University of Denmark, Richard Petersens Plads, Building 321, 2800 Kgs. Lyngby, Denmark*
[b]*Department of Transport, Technical University of Denmark, Bygningstorvet, Building 115, 2800 Kgs. Lyngby, Denmark*
[c]*Department of Management Engineering, Technical University of Denmark, Produktionstorvet, Building 426, 2800 Kgs. Lyngby, Denmark*

Abstract

Maritime transportation is the backbone of international trade. Over 80% of global merchandise trade is transported by sea. With an ever increasing volume of maritime freight, the efficient handling of both ships and containers has never been more critical. In this paper we consider the problem of allocating arriving ships to discrete berth locations at container terminals. This problem is recognized as one of the most important processes for any container terminal. We review and describe the three main models of the discrete dynamic berth allocation problem, improve the performance of one model, and, through extensive numerical tests, compare all models from a computational perspective. The results indicate that a generalized set-partitioning model outperforms all other existing models.

1. Introduction

Since the introduction of the container as we know it today in the early 50's the development in maritime transportation of cargo has been stunning. Since 1990, container trade is estimated to have increased by a factor of five (United Nations, UNCTAD Secretariat[2008]). The success is, to a large extent, based on the international standard size of a container. Containers are measured in multiples of 20 feet known as Twenty-foot Equivalent Units (TEUs). It is estimated that the global fleet of containers exceeds 23 million TEUs. In 2007, container cargo accounted for 1.24 billion of the 8.02 billion tonnes of all shipping cargo (United Nations, UNCTAD Secretariat[2008]), constituting an increase of 4.8% from the previous year.

Email addresses: katja.buhrkal@gmail.com (Katja Buhrkal), sarazuglian@yahoo.it (Sara Zuglian), sr@transport.dtu.dk (Stefan Ropke), jesla@man.dtu.dk (Jesper Larsen), rmlu@man.dtu.dk (Richard Lusby)

An important part of the global transportation of containers are the terminals where containers are loaded and/or unloaded. Here containers can change mode from sea to land (road or rail) and vice versa, or they can change from one vessel to another as a hub-and-spoke network is widely adopted. Large vessels can carry up to 15,000 TEUs and operate between huge transshipment terminals (hubs), while smaller vessels (so-called feeders) transport containers between smaller terminals (spokes) and the hubs. Globally it is estimated that the container terminals have a throughput of 485 million TEUs (United Nations, UNCTAD Secretariat[2008]) and that almost half of the container traffic in the world is handled by the 20 largest terminals.

A container terminal usually consists of three areas: the berth (where ships berth), the stowage area (where containers are stored temporarily) and the landside (where trucks and trains are serviced). The complexity of even medium sized terminals makes it impossible to consider the entire operation and plan it manually. Operations Research, therefore, has contributed to the planning and development within many of the terminal's processes. An overview of the different terminal operations and the impact of Operations Research are described in Steenken et al. [2004].

In this paper we focus on the Berth Allocation Problem (BAP). This problem entails assigning incoming ships to berth positions. Once a vessel is moored, it will remain at the berth until all required container processing has been completed. As berth space is very limited at most container terminals, and thousands of containers must be handled daily, an effective berth allocation is critical to the efficient management of the container flow. The BAP is recognized as one of the major container terminal optimization problems in Steenken et al. [2004], and it naturally lends itself towards a description in a two-dimensional space. One dimension is spatial, i.e. the quay length, while the other is a temporal decision horizon, which is often one week. Ships can be represented as rectangles whose dimensions are length and handling time. The handling time is defined to be the time the ship is at the berth, whereas the service time is the total time the ship spends at the port (i.e. the handling time plus any waiting time the ship experiences as a result of not being immediately serviced on arrival). These rectangles must be placed in the decision space without overlapping each other such that the length of the quay and the decision horizon are not violated (see Figure 1).

The handling time of a ship depends on its position at the quay. This reflects reality in that containers are prepared for particular ships, and the driving distances from the stowage area to the berth must be considered. As mentioned previously, the decision horizon is typically one week; however, this may be updated based on changes to arrival and departure times of ships.

BAP problems can be classified as being either *static* (SBAP) or *dynamic* (DBAP). The static case assumes that all ships are already in the port when the berth assignment is planned, while the latter allows for ships to arrive during container operations at the port. The BAP can be further classified into *discrete* and *continuous* variants. The discrete case allows only one ship at a time at each berthing location, regardless of its size, while the latter permits more.

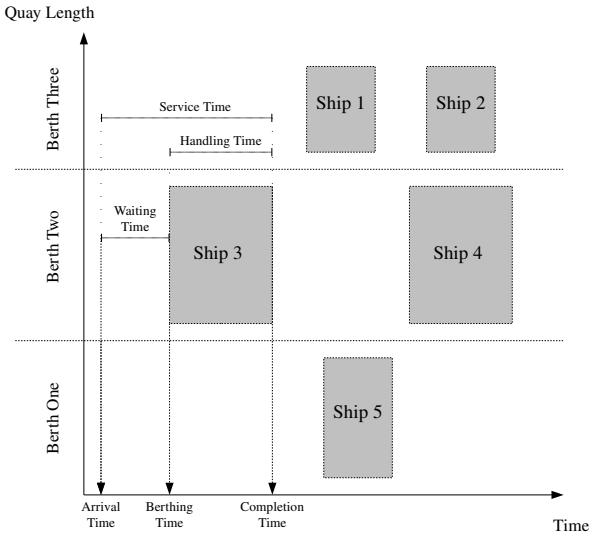


Figure 1: The representation of the berth-time space

This paper focuses on the discrete DBAP which is NP-Hard, see e.g. Monaco and Sammarra [2007]. We describe the three main models for this variant of the problem, improve the performance of one, and compare all models from a computational perspective. Hence, the contributions of this paper are threefold. Firstly, we show that the MDVRPTW model proposed in Cordeau et al. [2005], with a few improvements, is competitive with that proposed by Imai et al. [2001]. In Cordeau et al. [2005] it was concluded that the latter model is most efficient. Secondly, we show that a set-partitioning model proposed by Christensen and Holst [2008] is able to significantly outperform both aforementioned models. Finally, with the set-partitioning model we provide, for the first time, optimal solutions to all instances proposed in Cordeau et al. [2005]. This enables us to assess the quality of previously proposed heuristics.

The structure of this paper is as follows. We begin with a literature review in Section 2, while the different models for the discrete and dynamic variant of the BAP are presented in Section 3. A comparison of the models based on extensive computational experiments is presented in Section 4, and we conclude with a discussion on our findings in Section 5.

2. Literature Review

Studies on the BAP have appeared in the literature since the mid 1990's. The focus here is on the discrete DBAP. This is the most researched problem and, in what follows, we review the majority of work in this area. We briefly

deal with the other variants, as well as possible extensions, at the end of the review.

Imai et al. [2001] first proposed the DBAP. The ship handling time is assumed to be dependent on the assigned berth. Their model is an extension of that developed for the SBAP in Imai et al. [1997]. The Lagrangean Relaxation (LR) results in a linear assignment problem. In addition, three simple procedures are used to produce a feasible solution. Running times for the larger problems are terminated at 1500 seconds, and these still have considerable gaps to the lower bound. The instances contain at most 50 vessels and 10 berths. A similar approach is adopted in Monaco and Sammarra [2007]. A stronger LR is, however, given due to a reformulation of the problem. In Nishimura et al. [2001], a genetic algorithm is proposed. This approach divides the problem into subproblems based on time. Results are comparable in quality to Imai et al. [2001]; however, there is no comparison of running times for the two approaches.

Cordeau et al. [2005] model the problem as a Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). The objective function is to minimize the total ship service time. Furthermore, different positions along the berth lead to a different handling time for each vessel. With the MDVRPTW approach, ships are represented as customers, while berths are considered depots. A tabu search metaheuristic is then developed for the problem and tested on real-life instances. The MDVRPTW is also the basis of Mauri et al. [2008]. Their algorithm heuristically implements column generation. The master problem is a set-partitioning problem where each column is a sequence of vessels that use a given berth. The subproblem is solved using an evolutionary-based metaheuristic.

An alternative approach is presented in Christensen and Holst [2008]. Here the problem is formulated as a generalized set-partitioning problem. Time is discretized and for each berth and time interval a packing constraint ensures that at most one ship can be at the berth at any given time. A column in the model is a ship at a given berthing position in time and space. GUB constraints ensure that one column for each ship is selected. For small problems total enumeration is possible, for larger ones a branch-and-price approach is suggested.

In the SBAP, all ships are already in the port and it is merely a question of assigning positions at the quay to the ships. This problem was shown in Imai et al. [1997] to be solvable in polynomial time since it can be formulated as a linear assignment problem. The problem is initially stated as a multicriteria problem that minimizes overall handling time and dissatisfaction of the berthing order. The instances have up to 40 vessels and up to 5 berths. Dai et al. [2004] solve the continuous version of the SBAP as a rectangle packing problem with release time constraints. The authors implement a neighborhood search algorithm using simulated annealing. The problem is represented using sequence pairs (see Murata et al. [1995]), and running times are at most 650 s. High-quality solutions are achieved for light loads on the terminal. Furthermore, the paper also describes a simulation tool.

Several papers consider the length of a berth location to be continuous. A first approach is Lim [1998], and this uses a graph representation to solve the

problem as a restricted form of a two-dimensional packing problem. The edges must be assigned an orientation to determine the order of berthing. Experimental results are sparse. In Wang and Lim [2007], the approach is extended to a metaheuristic using a beam search framework. Results are reported based on 40 instances and suggest that the approach is state-of-the-art. Tong et al. [1999] is also based on Lim [1998]. The authors present an ant colony metaheuristic. Computational results are compared to a randomized first-fit heuristic and are not impressive.

In Guan and Cheung [2004] weights in the objective reflect the relative importance of vessels. Two mathematical models (a relative position formulation and position assignment formulation) are given, and an LR approach is used for the latter. A tree search is used to guide the search. Due to large running times, vessels are clustered and the tree search algorithm is then run on each cluster. Note that Guan et al. [2002] also describe a similar, but simpler, solution method for the continuous SBAP. It should be mentioned that Cordeau et al. [2005] also describe a continuous version of their tabu search metaheuristic. A heuristic for the continuous problem is also given in Imai et al. [2005].

Park and Kim [2002] model the continuous BAP as a mixed-integer linear program. The objective function minimizes the penalty cost associated with service delays and the handling cost that is incurred when placing a ship at a non-optimal location. The model ensures that the rectangles representing the ships do not overlap in both space and time. This is similar to the container loading and block layout from Onodera et al. [1991] and Chen et al. [1995]. However, the experimental results show that the computation time was too excessive for practical purposes. A grid is therefore introduced to approximate the continuous solution. This allows an efficient LR approach utilizing subgradient optimization to be adopted. A simple heuristic is used to obtain a feasible solution. The authors consider 50 realistic instances, and all are solved within 500 seconds. Due to the high running times, Kim and Moon [2003] present a simulated annealing heuristic based on exchanging the position of two rectangles. Solution times of up to 160 seconds are reported for the larger instances (up to 40 vessels). The authors state that the heuristic produces near optimal solutions.

The generalized set-partitioning approach of Christensen and Holst [2008] for the discrete case can intuitively be extended to the continuous case. Berth length is discretized into the required level of detail, and one has a packing constraint for each segment of the quay and time interval. Preliminary tests in Christensen and Holst [2008] show that the method suffers from symmetry and highly fractional LP relaxations. The authors also conclude that further research is necessary to make this model competitive.

A recent, popular extension of the BAP is to integrate it into the planning of quay cranes. Contributions include Park and Kim [2003], Imai et al. [2008], Liang et al. [2009], Meisel and Bierwirth [2009]. All approaches are based on heuristics. The method proposed by Park and Kim [2003] decomposes the problem into a BAP and a Crane Assignment Problem (CAP). The BAP is solved with an adaptation of the method from Park and Kim [2002], while

the CAP is solved using dynamic programming. Both Imai et al. [2008] and Liang et al. [2009] are based on genetic algorithms that build on a similar two-phase approach in which the crane scheduling is added to check for feasibility and re-scheduling of the cranes. In this sense Imai et al. [2008] is basically an extension of the genetic algorithm already presented in Nishimura et al. [2001]. Finally Meisel and Bierwirth [2009] try to use the metaheuristic squeaky wheel optimization and tabu search on the problem.

In a few other papers the BAP is extended with service priority. Service priority assigns a priority (or importance) to each ship. In Imai et al. [2003] this simply adds a term to the objective function to capture the different priorities. This extension is also added to the genetic algorithm of Nishimura et al. [2001], and a similar feature is included in the model proposed in Cordeau et al. [2005]. Contrastingly, the extension in Hansen et al. [2008] is more elaborate. Here each ship has both a handling time and a handling cost; however, in addition, a premium is included in the objective function for each ship that reflects early departure. The problem is solved with a variable neighborhood search.

3. Modeling the BAP

This section describes several mixed integer programming (MIP) models for the discrete and dynamic berth allocation problem. We begin, in Section 3.1, with a description of the first model proposed by Imai et al. [2001] and the extension DBAP+ proposed in Monaco and Sammarra [2007]. Section 3.2 describes a heterogeneous vehicle routing problem with time windows (HVRPTW) formulation which is based on the multi-depot vehicle routing problem with time windows (MDVRPTW) formulation proposed by Cordeau et al. [2005]. Section 3.3 describes a model that improves upon the HVRPTW formulation in order to make it easier to solve. Finally, in Section 3.4 we describe a generalized set-partitioning (GSPP) formulation that was originally proposed by Christensen and Holst [2008].

3.1. Imai et al. [2001] DBAP Formulation

The MIP model for the DBAP that is presented in Imai et al. [2001] is an extension of that which is proposed in Imai et al. [1997] for the static case. The decision variables govern the assignment of ships to berths as well as the order in which the ships will be processed at the berths. That is, each decision variable is a binary variable of the form x_{ip}^k which states that ship i will be serviced as the pth ship at berth k. To understand the model below, we introduce the following notation. Let us assume we have a set of berthing locations M (with $|M| = m$), a set of ships N (with $|N| = n$) that wish to berth, and a set of service orders P (with $|P| = n$). Further assume that the handling time spent by ship i at berth k is given by h_i^k, that the arrival time of ship i is given by a_i, and that s^k defines the time at which berth k becomes available for the berth allocation planning. The aforementioned parameters and sets are sufficient to formulate the SBAP (see Imai et al. [1997] for details); however, the following

additional information is needed for the DBAP. Unlike the SBAP, in the DBAP not all ships arrive at the assigned berth before s^k. Thus, the set W_k is needed to indicate the set of ships satisfying $a_i \geq s^k$. In addition, one must also define $P_{(p)} = \{q \in P : q < p\}$, which gives the set of service orders before p, and the decision variables y_{ip}^k which give the idle time at berth k between the departure of the $(p-1)$th ship and the arrival of the pth ship, if ship i is the pth ship to be serviced. The model in Imai et al. [2001] can then be stated as follows.

$$\min \quad \sum_{k\in M}\sum_{i\in N}\sum_{p\in P}\left\{(n-p+1)h_i^k + s^k - a_i\right\}x_{ip}^k + \sum_{k\in M}\sum_{i\in W_k}\sum_{p\in P}(n-p+1)y_{ip}^k \tag{1}$$

s.t.

$$\sum_{k\in M}\sum_{p\in P} x_{ip}^k = 1 \qquad \forall i \in N, \tag{2}$$

$$\sum_{i\in N} x_{ip}^k \leq 1 \qquad \forall k \in M,\ p \in P, \tag{3}$$

$$\sum_{l\in N}\sum_{q\in P_{(p)}}\left(h_l^k x_{lq}^k + y_{lq}^k\right) + y_{ip}^k \geq \left(a_i - s^k\right)x_{ip}^k \qquad \forall i \in W_k,\ p \in P,\ k \in M, \tag{4}$$

$$x_{ip}^k \in \{0,1\} \qquad \forall i \in N,\ p \in P,\ k \in M, \tag{5}$$

$$y_{ip}^k \geq 0 \qquad \forall i \in N,\ p \in P,\ k \in M. \tag{6}$$

The objective function, given by (1), minimizes the total waiting and handling times of every ship. Constraint sets (2) and (3) ensure that each ship is serviced at one berth and that each berth can service at most one ship at any time, respectively. Constraint set (4) (along with (6)) controls the idle time variables. Such a constraint states that for any ship i which arrives after berth k opens, the time at which it can start being serviced (given by the accumulated service times of all the ships preceding it on berth k and the respective idle times) must be no less than $a_i - s^k$. Obviously, if ship i is the pth ship serviced at berth k and its arrival time, a_i, is less than the completion time of the ship in position $p-1$, then $y_{ip}^k = 0$. Finally, constraints (5) enforce the binary integer restriction on the x_{ip}^k variables, and constraints (6) ensure that the idle time variables, y_{ip}^k, are non-negative. Cordeau et al. [2005] point out that this initial model neglects the fact that each berth $k \in M$ is likely to have a closing time e^k. The following additional set of constraints ensure that all ship servicing conducted at berth k falls within its respective time window $[s^k, e^k]$:

$$\sum_{i\in N}\sum_{p\in P}(h_i^k x_{ip}^k + y_{ip}^k) \leq e^k - s^k \qquad \forall k \in M. \tag{7}$$

Monaco and Sammarra [2007] show that the basic formulation in Imai et al. [2001] can be strengthened by realizing that the idle times are independent of which ship is eventually serviced in the pth position at berth k. Thus, the ship subscript can be dropped, and the idle time variables are restated as y_p^k, which simply gives the idle time between the start of the pth service and the completion of the $(p-1)$th service at berth k. Suppose $C_{(p-1)}^k$ gives the completion time of service $p-1$ at berth k, then the value of each y_p^k variables is given as:

$$y_p^k = \max\left\{0, \sum_{i \in N} a_i x_{ip}^k - C_{(p-1)}^k\right\} \qquad \forall k \in M,\ p \in P. \tag{8}$$

The authors also show how constraint set (8) can be linearized and further strengthened to the following:

$$\sum_{i \in W_k} (a_i - s^k) x_{ip}^k - \sum_{l \in P_{(p)}} \left(y_l^k + \sum_{j \in N} h_j^k x_{jl}^k \right) - y_p^k \leq 0 \quad \forall k \in M,\ p \in P. \tag{9}$$

The improved Imai et al. [2001] model is identical to the formulation provided above with the exception that the y_p^k variables are used instead of the y_{ip}^k variables, constraint set (4) is replaced with constraint set (9), and the objective function is rewritten using the new variables:

$$\begin{aligned} \min \quad & \sum_{k \in M} \sum_{i \in N} \sum_{p \in P} \left\{(n-p+1)h_i^k + s^k - a_i\right\} x_{ip}^k \\ & + \sum_{k \in M} \sum_{p \in P} (n-p+1) y_p^k \end{aligned} \tag{10}$$

The basic model (1)–(6) is denoted *DBAP*, while the improved formulation is denoted as *DBAP+* in the following. Both are tested in Section 4, where we compare the performance of the different formulations. Cordeau et al. [2005] observe that the objective function of this model cannot be decomposed into a weighted sum of ship service times and, for this reason, developed the following formulation.

3.2. Heterogeneous Vehicle Routing Problem With Time Windows Formulation

Cordeau et al. [2005] present a multi-depot vehicle routing problem with time windows (MDVRPTW) formulation of the BAP where berths correspond to depots, ships correspond to customers and a mooring sequence at a particular berth corresponds to a vehicle route (this was originally proposed in Legato et al. [2001]). We note that the notation can be simplified by viewing the problem as a heterogeneous vehicle routing problem with time windows (HVRPTW). By using the HVRPTW model one can define the problem on a graph instead of

a multigraph as in Cordeau et al. [2005]. The complexity of the problem does, however, remain unchanged.

In the following we use the same notation as in the previous section, unless otherwise stated. The HVRPTW is defined on a graph $G = (V, A)$ where the set of vertices $V = N \cup \{o, d\}$ contains a vertex for each ship as well as vertices o and d which mark the origin and destination nodes for any route in the graph. The set of arcs is a subset of $V \times V$. Each ship $i \in N$ has a time window $[a_i, b_i]$ that indicates the time at which the ship first arrives at the port and the time by which its departure from the port must be ensured. For the origin and destination vertices, the time window $[s^k, e^k]$ depends on the vehicle k as berths can be available at different times. Each ship has a relative importance v_i (objective function weight) similar to the service priority approach of Imai et al. [2003], and handling times h_i^k that are dependent on the respective berth locations. The full HVRPTW formulation is given below. This model has two types of decision variables: the binary decision variables $x_{ij}^k, k \in M, (i, j) \in A$ and the continuous variables $T_i^k, i \in V, k \in M$. Each x_{ij}^k takes the value one if ship j immediately succeeds ship i at berth k and is zero otherwise. Each T_i^k gives the time at which ship i moors at berth k, or is equal to a_i if ship i does not use berth k. The variables T_o^k and T_d^k define the start and end time of activities at berth $k \in M$

$$\min \quad \sum_{i \in N} \sum_{k \in M} v_i \left(T_i^k - a_i + h_i^k \sum_{j \in N \cup \{d\}} x_{ij}^k \right) \tag{11}$$

s.t.

$$\sum_{k \in M} \sum_{j \in N \cup \{d\}} x_{ij}^k = 1 \qquad \forall i \in N \tag{12}$$

$$\sum_{j \in N \cup \{d\}} x_{o,j}^k = 1 \qquad \forall k \in M \tag{13}$$

$$\sum_{i \in N \cup \{o\}} x_{i,d}^k = 1 \qquad \forall k \in M \tag{14}$$

$$\sum_{j \in N \cup \{d\}} x_{ij}^k = \sum_{j \in N \cup \{o\}} x_{ji}^k \qquad \forall k \in M,\ i \in N \tag{15}$$

$$T_i^k + h_i^k - T_j^k \leq (1 - x_{ij}^k) M_{ij}^k \qquad \forall k \in M,\ (i,j) \in A \tag{16}$$

$$a_i \leq T_i^k \qquad \forall k \in M,\ i \in N \tag{17}$$

$$T_i^k + h_i^k \sum_{j \in N \cup \{d\}} x_{ij}^k \leq b_i \qquad \forall k \in M,\ i \in N \tag{18}$$

$$s^k \leq T_o^k \qquad \forall k \in M \tag{19}$$

$$T_d^k \leq e^k \qquad \forall k \in M \tag{20}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall k \in M,\ (i,j) \in A \tag{21}$$

$$T_i^k \in \mathbb{R}^+ \qquad \forall k \in M,\ i \in V \tag{22}$$

The objective function (11) minimizes the weighted sum of ship service times. Constraint set (12) states that each ship must be assigned to exactly one berth k, while constraints (13) and (14) guarantee that for each berth k the degree of origin (resp. destination) nodes is one. Flow conservation for the remaining vertices is ensured by constraints (15). Consistency for berthing time and mooring sequence on each berth is achieved with constraints (16). Note that the constants M_{ij}^k are defined as $M_{ij}^k = max\{b_i + h_i^k - a_j, 0\}$, where $k \in M$, $(i,j) \in A$. Constraints (17) and (18) enforce the time window requirements for each ship. The berth availability time windows are enforced by constraints (19) and (20). Finally, constraints (21) and (22) define the respective domains of the decision variables. In the data sets used in Cordeau et al. [2005] there are some infeasible assignments of ships to berths. Let $I \subseteq N \times M$ be a set containing all infeasible ships $\times$ berth allocations. That is, if $(i,k) \in I$ then it is infeasible to assign ship i to berth k. Such infeasibilities can be handled by adding the constraint

$$\sum_{(i,k) \in I} \sum_{j \in N} (x_{ij}^k + x_{ji}^k) = 0$$

The model presented in this section is denoted *HVRPTW* in the following.

3.3. Improved HVRPTW Model

In this section we modify the model proposed in Section 3.2 with the aim of reducing computation time. To improve the model we consider breaking symmetries (Section 3.3.1), variable fixing (Section 3.3.2), and adding valid inequalities (Section 3.3.3). We denote the model presented in this section *HVRPTW+*.

3.3.1. Berth Symmetry

In the data sets used in Cordeau et al. [2005] several berths are identical in terms of their availability time window and the handling times for all ships. One also expects that identical berths would be present in real life data. Identical berths introduce many equivalent solutions, which makes solving the model difficult. Therefore, we propose changing the model in such a way that it deals with berth types rather than individual berths. Let us now denote the set of berth types as M and define $\beta^k, k \in M$ to be the number of berths of type k in the problem instance. It is an easy preprocessing step to determine the set of berth types given the set of berths. We hence replace constraints (13) and (14) with the following:

$$\sum_{j \in W} x^k_{o,j} = \beta^k \qquad \forall k \in M \tag{23}$$

$$\sum_{i \in N \cup \{o\}} x^k_{i,d} = \beta^k \qquad \forall k \in M \tag{24}$$

Constraints (23) and (24) allow the origin (destination) node for each berth type k to have as many outgoing (incoming) arcs as there are berths of type k. Since berths can be left unused, one must also change the domain of the variable representing the arc from origin to destination node for berths type with multiplicity greater than one:

$$x^k_{o,d} \in \{0, ..., \beta^k\} \quad \forall k \in M \tag{25}$$

We expect this change to improve the model's performance as soon as an instance contains some identical berths; both symmetry and the number of decision variables are reduced.

3.3.2. Variable Fixing

It is well known from the VRPTW literature that an arc (i, j) can be removed from the instance if the time windows make it impossible to serve node j directly after node i. Such arc removals are valid because one can guarantee that the arc cannot be part of a feasible solution. For the the HVRPTW model for the BAP we do not expect such reductions to be effective as the time windows considered

are very wide. For the same reason we do not expect time window reduction techniques like those proposed in Desrochers et al. [1992] to be effective.

Instead we propose to fix variables that can be part of a feasible solution. One can fix a variable x_{ij}^k if one can guarantee that an optimal solution exists in which berth type k does not use the arc (i, j). It is possible to fix x_{ij}^k to zero if variable x_{ji}^k is not fixed and $h_i^k \geq h_j^k \wedge a_i \geq a_j \wedge b_i \geq b_j$. To see that it is valid to fix the variable, consider a solution s_1 that uses arc (i, j) in a route for berth type k and assume that $h_i^k \geq h_j^k \wedge a_i \geq a_j \wedge b_i \geq b_j$. In this case, one can obtain a solution s_2 that is at least as good as s_1 by exchanging ship i and j such that ship j is served right before i. This is because 1) the service time of ship j gets reduced by at least h_i^k as ship j can start as early as ship i can, 2) the service time of ship i gets increased by at most h_j^k, 3) in solution s_2, ship i finishes earlier or, at worst, the same time as ship j did in solution s_1 (as $a_i \geq a_j$). Observations 1) and 2) ensure that the objective does not increase when looking at ship i and j as $h_i^k \geq h_j^k$, while observation 3) ensures that the ships following ships i and j on berth k do not get delayed and thereby increase the objective value. Furthermore, solution s_2 is feasible with respect to the end of the time windows due to observation 3 and the assumption that $b_i \geq b_j$.

Figure 2 shows an example. Ship j has an earlier and shorter time window than ship i. The handling time of ship j is also shorter than that of ship i. In solution s_1, ship i is served just before ship j and both are on berth k. The service time of ship i is only the handling time h_i^k, but ship j has to wait until ship i arrives and is served before the handling can begin. If the ships are exchanged solution, s_2 is obtained. The service time of ship j is reduced to the handling time, while that of ship i is only increased by the handling time of ship j minus the time until ship i arrives. Solution s_2 is clearly better than s_1.

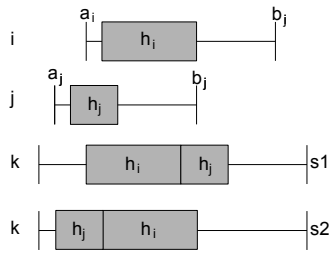


Figure 2: *The variable fixing*

One may wonder if it could be problematic to remove an arc (i, j) if the arc (h, i) is present for some $h \in V$ while (h, j) is removed by the arc reduction. This implies that the partial path consisting of the nodes $h \rightarrow i \rightarrow j$ would be feasible while $h \rightarrow j \rightarrow i$ would not. One can observe that, in this case, the partial path $j \rightarrow h \rightarrow i$ would be feasible because the arc reduction never removes arcs in both directions for a pair of nodes. The partial path $j \rightarrow h \rightarrow i$ would also be at least as good as $h \rightarrow i \rightarrow j$ since (h, j) is only removed if the

solution where h and j are exchanged is at least as good.

3.3.3. Valid Inequalities

We formulate a class of valid inequalities that aim to increase the lower bound of the T_i^k variables. Increasing these bounds will, most likely, increase the lower bound obtained from the solution to the LP relaxation as the T_i^k variables play an important role in the objective function. The valid inequality is:

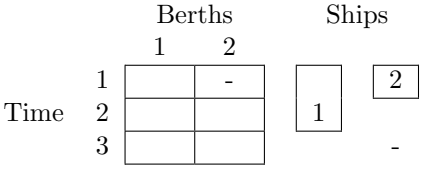
$$s^k x_{o,j} + \sum_{i \in N} \left(\max \left\{ a_i, s^k \right\} + h_i^k \right) x_{ij}^k \leq T_j^k \qquad \forall j \in N, k \in M \tag{26}$$

To see that the inequality is valid, first observe that at most one of the x variables on the left hand side can be 1 in a feasible solution. This is because of constraints (12) and (15). Having realized this, it is easy to see that the inequality is valid no matter which one of the x variables on the left hand side is non-zero. Note that we need the $\max \left\{ a_i, s^k \right\}$ expression since a_i may be smaller than s^k. The inequality is equivalent to a valid inequality proposed for the VRPTW proposed by Desrochers and Laporte [1991].

There are only $|N| \times |M|$ constraints of this type and these can be added to the formulation in advance. However, it should be noted that (26) uses the berth opening time and the arrival times of the ships. Therefore, it is most effective for the first couple of ships at each berth. In other words, we expect the inequalities to be most useful when the ratio of ships to berths is small.

3.4. Generalized Set-Partitioning

The BAP can also be modelled as a generalized set-partitioning problem (GSPP). This model was proposed by Christensen and Holst [2008]. The model assumes that all time measurements are integers. In order to illustrate how the constraint matrix is composed, a small example is provide below. This example has 2 berths, number 1 is open from time 1-3, and number 2 from 2-3. Ships 1 and 2 have handling time 2 and 1, respectively (on both berths) and ship number 2 must leave before time 3.



In the GSSP model, a column (variable) represents a feasible assignment of a single ship to a berth at a specific time. The first n rows correspond to the n ships. If a column represents an assignment of ship i, there will be a 1 in row i and zeros in the rest of the n first rows. Furthermore, there is one row for each available time unit at each berth. An entry in a column is equal to one if the ship occupies the berth at the considered time unit, otherwise it is zero. The cost of each column is equal to the service time arising from

the ship/berth/time assignment. The full column matrix corresponding to the example is given below:

	x_1	x_2	x_3	x_4	x_5	x_6
Cost	2	3	3	1	2	2
ship 1	1	1	1			
ship 2				1	1	1
berth1/time1	1			1		
berth1/time2	1	1			1	
berth1/time3		1				
berth2/time2			1			1
berth2/time3			1			

We now define the model more formally. The set of columns is denoted by Ω. We define two matrices A and B, both containing $|\Omega|$ columns. Matrix $A = (A_{i\omega})$ contains a row for each ship, and $A_{i\omega} = 1$ if and only if column ω represents an assignment of ship $i \in N$. Each column of A contains exactly one non-zero element. Matrix $B = (B_{p\omega})$ contains a row per (berth,time) position. The rows of B are indexed by the set P, with $|P| = \sum_{k \in M}(e^k - s^k)$. The entry $B_{p\omega}$ is one if and only if position $p \in P$ is contained in the assignment that column ω represents. The cost c_ω of any column $\omega \in \Omega$ is the service time of the respective position assignment and can be multiplied by the priority factor v_i if necessary. A binary variable x_ω is equal to one if column ω is used in the solution, and is zero otherwise. With these definitions we can present the GSPP formulation of the BAP.

$$\min \quad \sum_{\omega \in \Omega} c_\omega x_\omega \tag{27}$$

s.t.

$$\sum_{\omega \in \Omega} A_{i\omega} x_\omega = 1 \qquad \forall i \in N \tag{28}$$

$$\sum_{\omega \in \Omega} B_{p\omega} x_\omega \leq 1 \qquad \forall p \in P \tag{29}$$

$$x_\omega \in \{0, 1\} \qquad \forall \omega \in \Omega \tag{30}$$

The objective function minimizes the ships service time. Constraints (28) ensure that all ships are served, while constraints (29) enforce the restriction that only one ship can use any berth during each time interval. It is straightforward to write a program that generates the entire constraint matrix.

The GSPP is in general NP-hard. However, constraints (28) act as so-called generalized upper bound (GUB) constraints. The addition of which makes each of the submatrices containing only the columns of a specific ship perfect. This property guarantees that all extreme points of the feasible region (defined by the constraint system and the columns of any submatrix) are integer solutions.

When putting the submatrices together, however, this property is lost. In most cases, the addition of the GUB constraints makes the problem easier and often gives integer or less fractional solutions (see Ryan and Foster [1981] and Padberg [1974]).

In the following the models DBAP, DBAP+, HVRPTW and HVRPTW+ are denoted *compact* models while the GSPP model is denoted an *extensive* model. The reason for this vocabulary is the different growth of number of variables and constraints as a function of the instance size.

The GSPP model offers some modeling advantages compared to the compact models. It is easy to incorporate advanced constraints on the placement of ships as long as the constraints only consider a single ship because such constraints can be handled while generating the columns. For similar reasons it is possible to handle complicated objective functions as long as the objective function can be expressed as a sum of column costs. On the other hand, it is not easy to handle constraints that involves several ships in the GSPP model. It would for example be problematic to model a problem where the handling time of a ship is dependent on which ship that was served immediately before, on the same berth.

4. Computational Results

The models presented in Section 3 are compared in this section. For the comparison of the compact models five instance sizes were considered: 25 ships with 5, 7, and 10 berths; 35 ships with 7 and 10 berths. A set of 10 instances were generated for each. These correspond to the $I2$ set from Cordeau et al. [2005]. We also compare the GSPP model with the best heuristic methods from the literature (specifically, T^2S from Cordeau et al. [2005] and PTA/LP from Mauri et al. [2008]). For this, the $I3$ set from Cordeau et al. [2005] was used. The data consists of 30 instances and contains up to 60 ships and 13 berths. In all instances we have $v_i = 1$ for all ships $i \in N$.

The three compact models from the literature: DBAP, DBAP+ and HVRPTW as well as HVRPTW+ were implemented in OPL Studio. The greedy algorithm described in Section 4.1 was used in order to compute an initial solution for all of them. All tests were run on an Intel Xeon 5430 (2.66 GHz) processor and used a 32-bit version of CPLEX 11. The time limit for the solver was 2 hours and all computation times are given in seconds. We have used the standard parameters of CPLEX in all experiments. Section 4.2 compares the four models. The GSPP formulation from Section 3.4 was implemented by generating all columns a priori using a JAVA program and solving the resulting IP using CPLEX 11. In Section 4.3 we demonstrate that the GSPP model is superior to the four compact models and detailed comparisons with the two best compact models (DBAP+ and HVRPTW+) and the best heuristics from the literature are also provided.

4.1. Greedy Heuristic

A simple greedy heuristic was implemented in order to compute an upper bound for the BAP. It assumes that all berths become available at the same time and sorts the ships by arrival time. Then, it loops through the time and looks at all the available ships. It assigns ships one at a time by choosing the combination of berth and ship that will end first. The time is increased when there are no ships available or all berths are busy. This continues until all ships have been assigned, or one has reached a time point where all berths are closed, in which case the remaining ships cannot be serviced.

4.2. Computational Results for the DBAP, DBAP+, HVRPTW and HVRPTW+ Models

In this section the four compact models from Sections 3.1 to 3.3 are compared. The results can be seen in Table 1 and Table 2. Comparing DBAP and DBAP+ one can observe that the DBAP+ always provides a significant improvement over DBAP. This confirms the results in Monaco and Sammarra [2007]. Even though the CPLEX solver runs out of memory for many instances, the solution values that it calculates before running out of memory are far better than the ones that the DBAP formulation provided within the time limit. It may be possible to avoid some of the out-of-memory situations by tweaking the parameters of CPLEX but we have decided not to do so to enable a clean and fair comparison of the different models. Observe that the DBAP+ model can solve 12 out 25 instances to optimality.

For the HVRPTW and HVRPTW+ models, one can observe that HVRPTW+ always provides a significant improvement. HVRPTW+ performs much better as the ships/berths ratio decreases. This is due to the type of valid inequalities introduced; they mainly tighten the formulation for the first couple of ships at each berth (see Section 3.3.2). Observe that the HVRPTW+ model can solve 14 instances to optimality.

In Cordeau et al. [2005, p. 531] it was concluded that, from a computational point of view, DBAP is better than MDVRPTW (HVRPTW) in that it can solve larger instances. We have shown here that the MDVRPTW model proposed in Cordeau et al. [2005], with a few improvements, is competitive with the DBAP model proposed in Imai et al. [2001]. In fact, the HVRPTW+ formulation provides a much better optimality gap in most cases and solves more instances to optimality. However, the DBAP provides good upper bounds.

Based on the experiments we cannot conclude that DBAP+ dominates HVRPTW+ or vice versa. When comparing the DBAP+ and HVRPTW+ formulations it can be seen that the latter solves to optimality two new instances with respect to the DBAP+. In general, the DBAP+ formulation takes much more time than the HVRPTW+ (average on the same 12 instances: DBAP+ 1539 seconds and HVRPTW+ 337 seconds). However, in general DBAP+ provides the best upper bounds, especially for those instances with higher ships/berths ratio.

Instance		Greedy	DBAP			DBAP+			HVRPTW			HVRPTW+			Impr.[b]
		Value	Time[a]	Value	Gap[c]	Time[a]	Value	Gap[c]	Time[a]	Value	Gap[c]	Time[a]	Value	Gap[c]	(%)
25×5	01	868	-	801	27,68	5936*	759	3,54	-	860	41,63	-	762	3,66	-0,39
	02	1126	-	1030	36,73	2476*	955	11,59	-	1120	51,25	-	1045	23,35	-8,61
	03	1070	-	1002	10,53	1636*	970	3,79	-	1038	50,96	-	1010	25,64	-3,96
	04	813	-	729	75,11	2117*	688	10,76	-	749	38,32	-	701	9,69	-1,85
	05	1119	-	1060	33,94	3295*	958	5,83	-	1045	42,93	-	1001	12,78	-4,30
	06	1254	-	1139	27,08	2076*	1135	11,99	-	1152	51,2	-	1169	22,93	-2,91
	07	751	-	886	52,32	2702*	844	4,08	-	969	44,24	-	838	2,70	0,72
	08	751	-	669	76,70	2870*	628	3,18	-	722	38,78	-	627	0,39	0,16
	09	855	-	800	62,30	2101*	758	2,90	-	851	40,3	-	752	0,55	0,80
	10	1191	-	1144	26,15	3729*	1081	9,04	-	1191	51,81	-	1174	26,83	-7,92
Mean					**42,85**			**6,67**			**45,14**			**10,08**	**-2,83**
25×7	01	725	-	677	80,40	701	657		-	717	31,8	189	657		
	02	720	-	670	93,84	899	662		-	681	28,61	192	662		
	03	941	-	831	52,96	3003*	808	1,24	-	849	35,68	-	812	2,43	-0,49
	04	729	-	670	98,57	1791*	648	3,09	-	729	39,09	-	648	2,05	
	05	863	-	745	80,02	3252*	725	1,52	-	803	35,74	4348	725		
	06	927	-	883	100,00	1854*	795	12,83	-	901	43,84	-	800	9,14	-0,63
	07	826	-	801	100,00	2405*	745	6,85	-	826	37,53	-	736	2,50	1,22
	08	868	-	797	55,17	2228*	787	8,24	-	864	41,44	-	805	10,16	-2,24
	09	835	-	761	72,72	7196	749		-	835	38,2	690	749		
	10	933	-	875	86,37	2295	825		-	866	33,72	395	825		
Mean					**82,01**			**3,38**			**36,57**			**2,63**	**-0,21**
25×10	01	812	-	720	85,72	510	713		-	766	34,98	166	713		
	02	780	-	751	100,00	886	727		-	747	34,34	730	727		
	03	844	-	814	100,00	1182	761		-	784	29,59	178	761		
	04	886	-	851	90,75	1726*	814	3,23	-	884	39,82	-	810	1,47	0,49
	05	967	-	859	78,78	373	840		-	947	39,6	926	840		
	06	734	-	728	100,00	1059	689		-	734	32,96	101	689		
	07	720	-	703	100,00	802	666		-	709	33,71	170	666		
	08	950	-	918	100,00	1641	855		-	879	34,7	147	855		
	09	769	-	745	100,00	2563*	713	5,92	-	725	32,27	173	710		0,42
	10	860	-	807	100,00	920	801		-	821	33,01	160	801		
Mean					**95,53**			**0,92**			**34,50**			**0,15**	**0,09**

[a] A "-" indicates that the time limit of 7200 seconds was reached, a "*" indicates that the CPLEX solver ran out of memory.
[b] Comparison between the solutions provided by DBAP+ and HVRPTW+, calculated as $(DBAP+-HVRPTW+)/HVRPTW+$.
[c] The gap is calculated with respect to the value of the linear relaxation as (upper bound-lower bound)/upper bound.

Table 1: Computational results for DBAP, DBAP+, HVRPTW and HVRPTW+, Part I - 25 ships.

Instance		Greedy	DBAP			DBAP+			HVRPTW			HVRPTW+			Impr.[b]
		Value	Time[a]	Value	Gap[c]	Time[a]	Value	Gap[c]	Time[a]	Value	Gap[c]	Time[a]	Value	Gap[c]	(%)
35×7	01	1131	-	1037	29,24	-	1021	5,68	-	1131	41,29	-	1075	10,43	-5,02
	02	1387	-	1343	60,84	-	1237	9,62	-	1331	45,15	-	1357	17,91	-8,84
	03	1363	-	1309	45,65	-	1219	10,44	-	1353	47,38	-	1295	18,92	-5,87
	04	1306	-	1202	45,47	6226*	1176	8,76	-	1306	43,95	-	1306	17,61	-9,95
	05	1253	-	1253	65,42	-	1183	11,14	-	1253	43,5	-	1236	14,48	-4,29
	06	1871	-	1771	31,94	4739*	1701	6,91	-	1847	56,69	-	1849	30,67	-8,00
	07	1295	-	1206	61,93	5891*	1193	10,23	-	1295	45,76	-	1266	13,47	-5,77
	08	1512	-	1401	44,30	-	1334	10,72	-	1512	50	-	1437	19,00	-7,17
	09	1482	-	1482	77,62	5990*	1278	13,85	-	1482	50,33	-	1375	18,48	-7,05
	10	1318	-	1203	50,48	5611*	1159	8,66	-	1318	46,28	-	1262	16,82	-8,16
Mean					**51,29**			**9,60**			**47,03**			**17,78**	**-7,01**
35×10	01	1240	-	1220	87,30	5311*	1137	3,25	-	1206	39,62	-	1124	1,87	1,16
	02	1360	-	1304	57,75	-	1209	6,95	-	1360	46,32	-	1271	11,48	-4,88
	03	1090	-	1090	100,00	1663*	945	2,22	-	1033	37,46	-	947	2,22	-0,21
	04	1370	-	1370	89,28	5938*	1288	8,07	-	1370	43,63	-	1297	8,13	-0,69
	05	1491	-	1491	54,83	6352*	1368	5,04	-	1491	43,66	-	1388	6,99	-1,44
	06	1431	-	1428	72,74	6008*	1223	5,40	-	1365	42,86	-	1316	11,78	-7,07
	07	1190	-	1190	76,81	3346*	1074	3,39	-	1190	41,22	-	1068	2,59	0,56
	08	1311	-	1311	100,00	3918*	1216	7,57	-	1246	41,57	-	1217	6,77	-0,08
	09	1410	-	1359	31,42	-	1326	10,76	-	1410	47,52	-	1388	16,57	-4,47
	10	1350,00	-	1311	78,28	5595*	1191	2,02	-	1318	41,27	-	1192	0,84	-0,08
Mean					**74,84**			**5,47**			**42,51**			**6,92**	**-1,72**

[a] A "-" indicates that the time limit of 7200 seconds was reached, a "*" indicates that the CPLEX solver ran out of memory.
[b] Comparison between the solutions provided by DBAP+ and HVRPTW+, calculated as $(DBAP+ - HVRPTW+)/HVRPTW+$.
[c] The gap is calculated with respect to the value of the linear relaxation as (upper bound-lower bound)/upper bound.

Table 2: Computational results for DBAP, DBAP+, HVRPTW and HVRPTW+, Part II - 35 ships.

4.3. GSPP Model Results

The best compact models, DBAP+ and HVRPTW+, are compared, in Table 3, with the GSPP and the heuristic T^2S from Cordeau et al. [2005] on the instances containing 25 and 35 ships. For the compact models we only indicate whether the instances are solved to optimality ($\surd$), or not ($\div$). The time to generate all columns for the GSPP was always less than one tenth of a second and was not included in the computation time presented in Table 3. The percentage difference between the GSPP and T^2S is shown in the table: with an average difference of 1.0% and a worst case of 2.8%, we conclude that T^2S performs well within a few seconds. (Cordeau et al. [2005] is not more specific about running times for T^2S). However, the GSPP is also fast and guarantees optimality.

The computational results obtained on the bigger instances (60 ships and 13 berths) for the GSPP and the two heuristics (T^2S and PTA/LP) are compared in Table 4. The PTA/LP heuristic was executed on an AMD Athlon 64 3500 (2.2GHZ) which must be expected to be slower than our computer. It can be seen that PTA/LP dominates T^2S in terms of solution quality. In fact, PTA/LP is only 1 time unit away from optimality in 3 out of 30 cases. However, its runtime is slower than the GSPP, even when taking the different CPUs into account. It should be noted that Cordeau et al. [2005] report a solution of 1212 on instance i10 which is better than the optimal solution. Private correspondence with the authors revealed that this is due to a typo in Table 7 in Cordeau et al. [2005]. The solution found by their heuristic was actually the optimal one, with cost 1213.

5. Conclusion

In this paper we have reviewed and compared five different models for the discrete and dynamic berth allocation problem. We have demonstrated that the GSPP model is superior to all others on the set of instances from Cordeau et al. [2005]. The performance of the model is quite remarkable. For all the instances considered, the model finds the optimal solution within 30 seconds. This is in sharp contrast to the results in the previous state-of-the-art paper (Monaco and Sammarra [2007]), where none of the instances could be solved to optimality within two hours.

This paper also shows that the MDVRPTW/HVRPTW model proposed in Cordeau et al. [2005] is also quite attractive from a computational point of view. With a few, simple improvements the model is competitive with all other models except the GSPP model.

Instance		DBAP+	HVRPTW+	GSPP		T^{2}S	Difference[d]
		Solved	Solved	Time	Value	Value	(%)
25×5	01	÷	÷	5.99	759	759	
	02	÷	÷	3.70	964	965	0.1
	03	÷	÷	2.95	970	974	0.4
	04	÷	÷	2.72	688	702	2.0
	05	÷	÷	6.97	955	965	1.0
	06	÷	÷	3.10	1129	1129	
	07	÷	÷	2.31	835	835	
	08	÷	÷	1.92	627	629	0.3
	09	÷	÷	4.76	752	755	0.4
	10	÷	÷	6.38	1073	1077	0.4
Mean		**0**	**0**	**4.08**			**0.5**
25×7	01	√	√	3.62	657	667	1.5
	02	√	√	3.15	662	671	1.4
	03	÷	÷	4.28	807	823	2.0
	04	÷	÷	3.78	648	655	1.1
	05	÷	√	3.85	725	728	0.4
	06	÷	÷	3.60	794	794	
	07	÷	÷	3.54	734	740	0.8
	08	÷	÷	3.93	768	782	1.8
	09	√	√	3.73	749	759	1.3
	10	√	√	3.82	825	830	0.6
Mean		**4**	**5**	**3.73**			**1.1**
25×10	01	√	√	5.83	713	717	0.6
	02	√	√	6.99	727	736	1.2
	03	√	√	6.12	761	764	0.4
	04	÷	÷	5.38	810	819	1.1
	05	√	√	6.77	840	855	1.8
	06	√	√	5.57	689	694	0.7
	07	√	√	5.83	666	673	1.1
	08	√	√	5.87	855	860	0.6
	09	÷	√	5.38	711	726	2.1
	10	√	√	5.96	801	812	1.4
Mean		**8**	**9**	**5.97**			**1.1**
35×7	01	÷	÷	12.57	1000	1019	1.9
	02	÷	÷	15.93	1192	1196	0.3
	03	÷	÷	7.16	1201	1230	2.4
	04	÷	÷	13.59	1139	1150	1.0
	05	÷	÷	11.50	1164	1179	1.3
	06	÷	÷	29.16	1686	1703	1.0
	07	÷	÷	12.89	1176	1181	0.4
	08	÷	÷	17.52	1318	1330	0.9
	09	÷	÷	8.41	1245	1245	
	10	÷	÷	15.16	1109	1130	1.9
Mean		**0**	**0**	**14.39**			**1.1**
35×10	01	÷	÷	19.98	1124	1128	0.4
	02	÷	÷	11.37	1189	1197	0.7
	03	÷	÷	8.97	938	953	1.6
	04	÷	÷	10.28	1226	1239	1.1
	05	÷	÷	22.31	1349	1372	1.7
	06	÷	÷	10.92	1188	1221	2.8
	07	÷	÷	9.74	1051	1052	0.1
	08	÷	÷	9.39	1194	1219	2.1
	09	÷	÷	29.45	1311	1315	0.3
	10	÷	÷	10.36	1189	1198	0.8
Mean		**0**	**0**	**14.28**			**1.1**

[d] Comparison between the solutions provided by GSPP and T^{2}S, calculated as $(GSPP - T^2S)/GSPP$.

Table 3: Comparison between GSPP and T^{2}S - 25 and 35 ships instances.

Instance		GSPP		T^{2}S	PTA/LP		GSPP vs. PTA/LP[e]
		Time	Value	Value	Time	Value	(%)
60×13	**i01**	17.92	1409	1415	72.14	1409	
	i02	15.77	1261	1263	58.92	1261	
	i03	13.54	1129	1139	94.62	1129	
	i04	14.48	1302	1303	103.16	1302	
	i05	17.21	1207	1208	72.20	1207	
	i06	13.85	1261	1262	74.22	1261	
	i07	14.60	1279	1279	86.73	1279	
	i08	14.21	1299	1299	48.77	1299	
	i09	16.51	1444	1444	91.86	1444	
	i10	14.16	1213	1213	61.81	1213	
	i11	14.13	1368	1378	95.34	1369	0.07
	i12	15.60	1325	1325	77.39	1325	
	i13	13.87	1360	1360	62.55	1360	
	i14	15.60	1233	1233	69.05	1233	
	i15	13.52	1295	1295	71.28	1295	
	i16	13.68	1364	1375	169.81	1365	0.07
	i17	13.37	1283	1283	32.89	1283	
	i18	13.51	1345	1346	81.78	1345	
	i19	14.59	1367	1370	122.00	1367	
	i20	16.64	1328	1328	74.25	1328	
	i21	13.37	1341	1346	103.52	1341	
	i22	15.24	1326	1332	104.17	1326	
	i23	13.65	1266	1266	41.59	1266	
	i24	15.58	1260	1261	75.81	1260	
	i25	15.80	1376	1379	95.09	1376	
	i26	15.38	1318	1330	70.00	1318	
	i27	15.52	1261	1261	77.38	1261	
	i28	16.22	1359	1365	51.52	1360	0.07
	i29	15.30	1280	1282	196.36	1280	
	i30	16.52	1344	1351	69.62	1344	
Mean		**14.98**			**83.53**		**0.01**

[e] Comparison between the solutions provided by GSPP and PTA/LP, calculated as $(GSPP - PTA/LP)/GSPP$.

Table 4: Comparison between GSPP, T^{2}S and PTA/LP - 60 ships instances.

References

C.S. Chen, S.M. Lee, and Q.S. Shen. An analytical model for the container loading problem. *European Journal of Operational Research*, 80:68 –76, 1995.

C.G. Christensen and C.T. Holst. Berth allocation in container terminals. Master's thesis, Department of Informatics and Mathematical Modelling, Technical university of Denmark, April 2008. In Danish.

J.-F. Cordeau, G. Laporte, P. Legato, and L. Moccia. Models and tabu search heuristics for the berth-allocation problem. *Transportation Science*, 39(4): 526–538, 2005.

J. Dai, W. Lin, R. Moorthy, and C.-P. Teo. Berth allocation planning optimization in container terminal. Unpublished manuscript, June 2004.

M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10:27 – 36, 1991.

M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40 (2):342–354, 1992.

Y. Guan and R. K. Cheung. The berth allocation problem: models and solution methods. *OR Spectrum*, 26:75 – 92, 2004.

Y. Guan, W.-Q. Xiao, R.K. Cheung, and C.-L. Lee. A multiprocessor task scheduling model for berth allocation:heuristic and worst-case analysis. *Operations Research Letters*, 30:343 – 350, 2002.

P. Hansen, C. Oguz, and N. Mladenovic. Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, 191:636 – 649, 2008.

A. Imai, K. Nagaiwa, and C. W. Tat. Efficient planning of berth allocation for container terminals in asia. *Journal of Advanced Transportation*, 31(1):75 – 94, 1997.

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

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

A. Imai, X. Sun, E. Nishimura, and S. Papadimitriou. Berth allocation in a container port: using a continuous location space approach. *Transportation Science Part B*, 39:199 – 221, 2005.

A. Imai, H. C. Chen, E. Nishimura, and S. Papadimitriou. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E*, 44:900 – 920, 2008.

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

P. Legato, F. Monaco, and N. Tigani. Berth planning at gioia tauro's maritime terminal by logistic distribution models. *Annual Conf., AIRO, Cagliari, Itlay*, 2001.

C. Liang, Y. Huang, and Y. Yang. A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. *Computers & Industrial Engineering*, 56:1021 – 1028, 2009.

A. Lim. The berth planning problem. *Operations Research Letters*, 22:105 – 110, 1998.

G. R. Mauri, A. C. M. Oliveira, and L. A. N. Lorena. A hybrid column generation approach for the berth allocation problem. In J. van emert and C. Cotta, editors, *EvoCOP 2008*, volume 4972 of *Lecture Notes in Computer Science*, pages 110 – 122. Springer, 2008.

F. Meisel and C. Bierwirth. Heuristics for the integration of crane productivity in the erth allocation problem. *Transportation Research Part E*, 45:196 – 209, 2009.

M. F. Monaco and M. Sammarra. The berth allocation problem: A strong formulation solved by a lagrangean approach. *Transportation Science*, 41(2): 265 – 280, 2007.

H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. Rectangle-packing-based module placement. In *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*, pages 472 – 479. IEEE/ACM, November 1995.

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

H. Onodera, Y. Taniguche, and K. Tmaru. Branch-and-bound placement for building block layout. In *28th ACM/IEEE design automation conference*, pages 433 – 439, 1991.

M. W. Padberg. Perfect zero-one matrices. *Mathematical Programming*, 6:180 – 196, 1974.

K.T. Park and K.H. Kim. Berth scheduling for container terminals by using a sub-gradient optimization technique. *Journal of the Operational Research Society*, 53:1054 – 1062, 2002.

Y.-M. Park and K. H. Kim. A scheduling method for berth and quay cranes. *OR Spectrum*, 25:1 – 23, 2003.

D.M. Ryan and B.A. Foster. An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport*, pages 269–280. North-Holland Publishing Company, 1981.

D. Steenken, S. Voss, and R. Stahlbock. Container terminal operation and operations research – a classification and literature review. *OR Spectrum*, 26: 3 – 49, 2004.

C. J. Tong, H. C. Lau, and A. Lim. Ant colony optimization for the ship berthing problem. In P.S. Thiagarajan and R. Yap, editors, *ASIAN'99*, volume 1742 of *Lecture Notes in Computer Science*, pages 359 – 370. Springer, 1999.

United Nations, UNCTAD Secretariat. Review of maritime transport 2008, 2008. United Nations Publications, United Nations Conference on Trade and Development.

F. Wang and A. Lim. A stochastic beam search for the berth allocation problem. *Decision Support Systems*, 42:2186 – 2196, 2007.