ELSEVIER

Transportation Research Part A 38 (2004) 81–99

TRANSPORTATION
RESEARCH
PART A

www.elsevier.com/locate/tra

Stowing a containership: the master bay plan problem

Daniela Ambrosino, Anna Sciomachen *, Elena Tanfani

Dipartimento di Economia e Metodi Quantitativi (DIEM), Università di Genova, Via Vivaldi 2, 16126 Genova, Italy

Received 1 July 2002; received in revised form 19 September 2003; accepted 19 September 2003

Abstract

In this paper we are involved with the so-called master bay plan problem (MBPP), that is the problem of finding optimal plans for stowing containers into a containership, with respect to a set of structural and operational restrictions. We describe in detail such constraints and give a basic 0–1 Linear Programming model for MBPP. Successively, we present a heuristic approach that enables us to relax some relations from the model and give some prestowage rules for being able to solve this combinatorial optimization problem. In particular, we split the set of available locations of the ship into different subsets and force the stowage of containers within them depending on their features and handling operations.

A validation of the proposed approach is given together with the analysis of real instances of the problem coming from a maritime terminal located in the city of Genoa.

Keywords: Logistics; Maritime container terminal; Ship planning; Linear programming; Heuristic procedure

1. Introduction

The management of a terminal container is a complex process involving many interrelated decisions. Containers arrive and leave the terminal in various ways, for instance, by truck, train and vessel; the terminal is hence a basic node in the transportation network, and for this reason all the operations involved in the flow of containers have to be optimised in order to achieve maximum global productivity, expressed in terms of some appropriate economic indicators. Due to the high cost related to the total time spent by a ship in a terminal, all maritime companies refer to productivity indicators, e.g. the hourly container handling operations, for choosing the routes of their ships and the sequence of harbours to visit.

* Corresponding author.
E-mail address: sciomach@economia.unige.it (A. Sciomachen).

0965-8564/$ - see front matter
doi:10.1016/j.tra.2003.09.002

In this work we deal with the ship planning problem. The stowage of a containership is one of the problems that has to be solved on a daily basis by any company which manages a container terminal (Thomas, 1989). In the past, the stowage of containers was performed by the Captain of the ship; today, as a consequence of containerisation, the terminal has to decide the stowage of containers in accordance with the stowage instructions of the ship coordinator representing the maritime company which operates the ship.

The stowage of a containership involves different objectives; among others it is required to optimise the available space and prevent damage to the goods, the containership, its crew and its equipment. Moreover, it is desired to minimise the berthing time of the containership at the terminal (Atkins, 1991).

Formally, the master bay plan problem (MBPP) involves determining how to stow a set C of m containers of different types into a set S of n available locations within a containership, with respect to some structural and operational constraints related to both the containers and the ship, whilst minimising the total stowage time (see, e.g., Ambrosino and Sciomachen, 1998).

The problem we are investigating is really complicated because of its combinatorial nature. In the recent operations research and management science literature, decision support systems, heuristics, genetic algorithms, analytical and stochastic models have been suggested as very interesting approaches for solving problems that unfortunately have only some commonalties with MBPP and are mainly devoted to the loading problem (see Avriel and Penn, 1993; Avriel et al., 2000; Bischoff and Mariott, 1990; Bischoff and Ratcliff, 1995; Bortfeldt and Gehring, 2001; Chen et al., 1995; Crainic et al., 1993; Davies and Bischoff, 1999; Gehring et al., 1990; Gehring and Bortfeldt, 1997; Imai et al., 2002; Raidl, 1999; Wilson and Roach, 2000 among others).

Our first attempt to understand the problem in order to derive some rules for determining good container stowing plans has been reported in Ambrosino and Sciomachen (1998), where a constraints satisfaction approach has been used for defining and characterising the space of feasible solutions without having any objective function to optimise.

In this paper the main constraints of the problem are presented in detail in Section 2. A basic 0–1 Linear Programming model for MBPP is given in Section 3, while our proposed algorithmic approach is presented in Section 4. In Section 5 we give a simple test case and present some computational results aimed at validating the proposed approach. The study related to the problem of stowing containers in an Italian maritime terminal is given in Section 6. Finally, we give some comments and outlines for future work.

2. The structure of a containership and the constraints of the problem

When solving MBPP, of particular interest are the constraints related to the structure of the ship and the size of the hold and upper deck. We consider here two types of containerships, namely Ro–Ro (Roll on–Roll off), which load/unload containers through the ramps located either at the bow or the stern of the ship, and Lo–Lo (Lift on–Lift off), which load/unload containers from the top (by using cranes).

To give an idea of how stowage takes place, let us consider the basic structure of a ship (see Fig. 1) and its cross section. This consists of a given number of locations, that can vary in size depending on the ship. The most common, however is 8 feet (8′) in height, 8′ in width and 20′ in

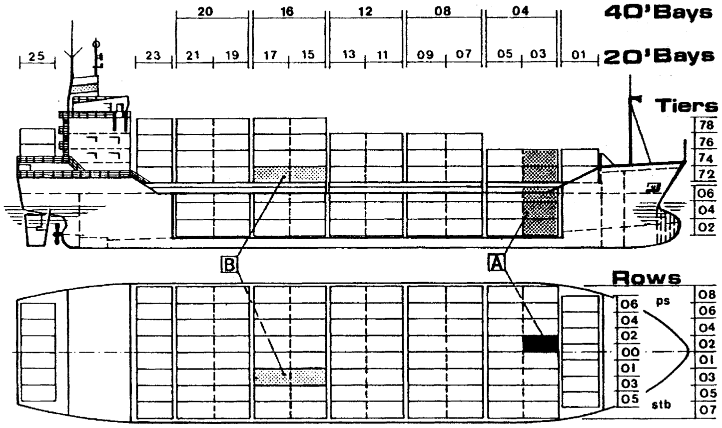

Fig. 1. Horizontal and cross sections of a standard containership.

length. Each location is identified by three indices, each one consisting of two numbers that give its position with respect to the three dimensions. In particular, each location is addressed by the following identifiers: (a) *bay*, that gives its position relative to the cross section of the ship (counted from bow to stern); (b) *row*, that gives its position relative to the vertical section of the corresponding bay (counted from the centre to outside); (c) *tier*, that gives its position related to the horizontal section of the corresponding bay (counted from the bottom to the top of the ship).

Thus a container is located in a given bay, on a given row and on a given tier. We will denote by I, J and K, respectively, the set of bays, rows and tiers of the ship. Moreover, we will use the following additional notation:

E and O: sets of even and odd bays, respectively, such that $E \subset I$, $O \subset I$ and $E \cup O \equiv I$;
A and P: sets of anterior and posterior bays, respectively, such that $A \subset I$, $P \subset I$ and $A \cup P \equiv I$;
R and L: sets of right side and left side rows, respectively, such that $R \subset J$, $L \subset J$ and $L \cup R \equiv J$.

Note that the address number of the ship locations depends on the system adopted by each maritime company. One of the most used, that is also chosen in this paper, ranks in an increasing order the first index, i.e. the bay, counted from bow to stern. In particular, each 20′ bay is numbered with an odd number, i.e. bay 01, 03, 05, etc., while two contiguous odd bays conventionally yield one even bay, for the stowage of 40′ containers, i.e. bay 04 = bay 03 + bay 05 (see Fig. 1). The effective even bays depend on the particular structure of the ship under consideration but, in any case, each even bay (for instance bay 04), is associated with two contiguous odd bays (for instance bay 03 and bay 05). Consequently, both the first and last bays will have an odd number. As for the second index, that is the row, ship locations have an even number if they are located on the seaside, i.e. row 02, 04, 06, and an odd number if they are located on the yard side, i.e. row 01, 03, 05, etc. Finally, for the third index, that is the tier, the levels are numbered from the bottom of the hold to the top with even number, i.e. tier 02, 04, 06, etc., while in the upper deck possible numbers are 82, 84 and 86. Note that, in the final stowage plan such tier numbers allow the containers stowed in the hold to be distinguished from those in the upper deck.

The number of bays, rows and tiers of each ship is known and together with the list containing all the characteristics of the containers to be loaded are the input data required for making explicit the constraints of the problem.

Together with the ship profile, that contains information related to both structural and operational constraints of the ship, the terminal has the bay plan configuration that is useful for defining the stowage in the available locations of the ship. An example of a bay plan configuration is given in Fig. 2.

Other useful information are provided by the ship coordinator. In particular, he/she gives instructions for the stowage by defining the available bays for containers having different destinations, and specifies the requirements for the location of reefers, hazardous and over sized containers.

Solving MBPP means taking into account both the constraints related to the particular ship under consideration and those of the containers.

Size of container. We consider here the standard size of a container, namely 20 and 40 feet in length with a section of $8' \times 8'$. Moreover, we refer to the container system expressed in terms of TEUs (Twenty-foot Equivalent Units); that is a TEU is 8 feet wide, 8 feet high and 20 feet long, and a 40 feet container is equivalent to two TEUs.

Containers of 40′ require two contiguous bow–stern locations of 20′ each, that is to say they have to be located in even bays (for instance bay 02). Consequently, the locations of the same row

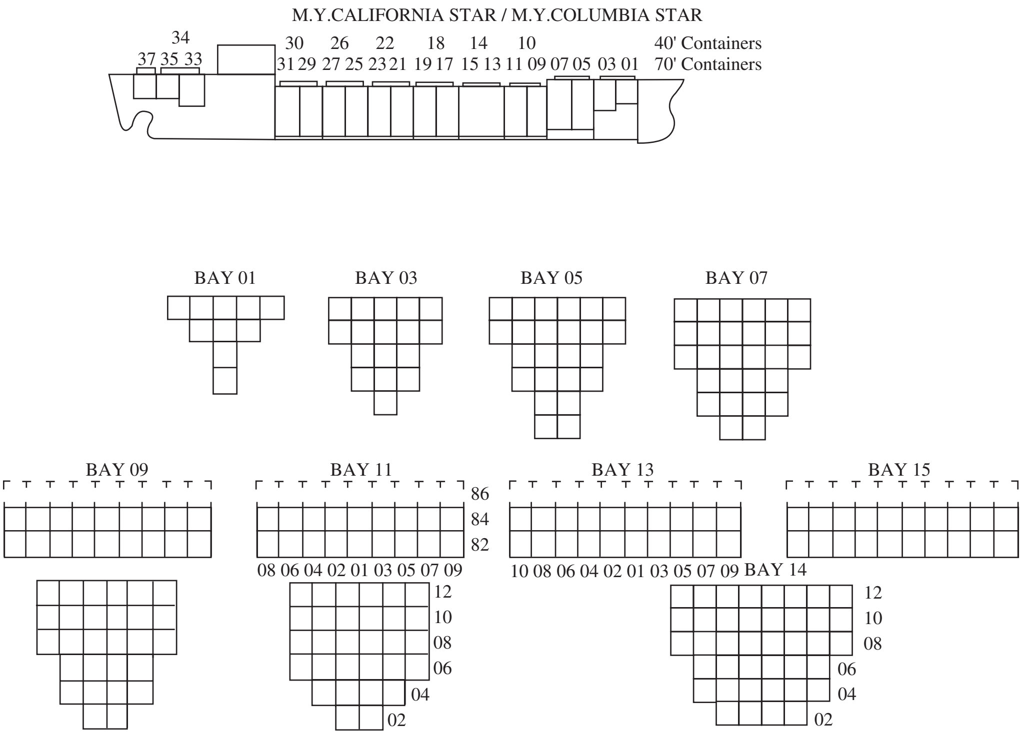

Fig. 2. Example of bay plan configuration.

and tier corresponding to two contiguous odd bays (e.g. bay 01 and bay 03) are not anymore available for stowing containers of 20′. Moreover, as it is always required for security reasons, we assume that 40′ containers cannot be located either under locations where 20′ containers are already stowed or over empty locations, while 20′ containers cannot be put over either 40′ containers or empty locations.

Type of container. Different types of container can usually be stowed in a containership, such as standard, carriageable, reefer, out of gauge and hazardous.

As it has been already said, the location of reefer containers is defined by the ship coordinator (who has a global vision of the trip), so that we know their exact position. This is generally near plugs in order to maintain the required temperature during transportation. Hazardous containers are also assigned by the harbour-master's office which authorises their loading. In particular, hazardous containers cannot be stowed either in the upper deck or in adjacent locations. They are considered in the same way as 40′ containers.

In the following we will denote the sets of standard 20 and 40 feet containers as T and F respectively, while H will denote the set of both reefer and hazardous containers to load on board.

Weight of container. The standard weight of an empty container ranges from 2 to 3.5 tons, while the maximum weight of a full container to be stowed in a containership ranges from 20 to 32 and 30 to 48 tons for 20′ and 40′ containers respectively. Containers are put in the yard into different stacks on the basis of their size, destination, type and weight class. This allows a priori identification of those containers that have to be located in the hold and those that have to be located in the upper deck. The total weight of all containers cannot exceed the maximum weight capacity, say Q, of the containership.

Usually, three classes of weight are considered, namely light (from 5 to 15 tons), medium (from 15 to 25 tons) and heavy containers (more than 25 tons). In this work, we will denote by w_c the weight of container c, $c = 1, \ldots, m$, that could belong to either class lw (for light weight), or mw (for medium weight) or hw (for heavy weight).

Moreover, the weight of a stack of three containers of 20′ and 40′ cannot be greater than an a priori established value, say MT and MF respectively; finally, the weight of a container located in a tier cannot be greater than the weight of the container located in a lower tier having the same row and bay.

Destination of container. A good general stowing rule suggests to load first (i.e. in the lower tiers) those containers having as destination the final stop of the ship and load last those containers that have to be unloaded first. Moreover, in the case of Ro–Ro ship, it is necessary to consider that in each odd bay there are stern ramps that allow entry to the hold from the upper deck. This implies that we cannot load in the hold containers to be unloaded before those located in the upper deck near the hatchcovers.

In Section 4 we will see that we can relax some of these constraints and use a heuristic pre-stowing procedure to assign a priori groups of bays to containers according to their destination.

In the following we will denote by $D = \{1, 2, \ldots, p\}$ the set of possible destinations and by $d_c \in D$ the destination of container c.

Distribution of container. Such constraints are related to a proper weight distribution in the ship. This is the basic condition for good stowage. Note that it is possible to check the stability of the ship by using some mathematical expressions only when the ship is loaded. However, there are some rules, mainly derived from experience and from the knowledge of the load, that help in establishing the most suitable way for stowing containers that assures a well balanced distribution of the weight. In practice, the most heavy containers are located in the hold, while the others are located in the upper deck. Moreover, for safety reasons, after the loading operation is complete, we have to verify different kinds of equilibrium, namely: (a) cross equilibrium, that is the weight on the right side of the ship, including the odd rows of the hold and upper deck, must be equal (within a given tolerance, say Q_1) to the weight on the left side of the ship, including the even rows of the hold and upper deck; (b) horizontal equilibrium, that is the weight on the stern must be equal (within a given tolerance, say Q_2) to the weight on the bow; (c) vertical equilibrium, that is the weight on each tier must be greater or equal than the weight on the tier immediately over it.

Note that a ship must be loaded in such a way that it remains able to travel independently in a variety of the weather conditions and that the stability constraints must also be satisfied after some possible loading/unloading at intermediate destinations.

In the next section we present a model that assumes all containers are ready to be loaded on the quay without considering their stock position in the yard. Work is in progress by the authors aimed at also including yard constraints and thus reflecting the behaviour of the planning office which makes the stowage plans in accordance with the stocking area requirements and the picking list for the containers in the yard. Interesting problems related to the organization of the stocking area have been considered in Kim et al. (2000), Preston and Kozan (2001) and Taleb-Ibraimi et al. (1993).

3. The basic model of the MBP problem

We now introduce our 0–1 model of this combinatorial optimization problem, that is the basis of the proposed method.

We assume x_{lc}, $l = 1, \dots, n$, $c = 1, \dots, m$, as decision variables of the problem, with the following specification:

$$x_{lc} = \begin{cases} 1 & \text{if a container } c \text{ is stowed in location } l \\ 0 & \text{otherwise} \end{cases}$$

Note that the lth location is actually identified by indices i, j, k representing, respectively, its bay, row and tier address, while c identifies the number (or code) of the cth stowed container. This means that, in practice, variable $x_{lc} = x_{ijkc}$, directly giving the location where container c is stowed if it is set to 1. Therefore, at the optimal solution we have the exact position of each container in the ship.

The definition of variable x_{lc}, $\forall l \in S$, $\forall c \in C$, enables an easy formulation of the underlying model for MBPP, reported here below.

Model MBPP:

$$\text{Min } L = \sum_l \sum_c t_{lc} x_{lc} \tag{1}$$

$$\sum_l \sum_c x_{lc} = m \tag{2}$$

$$\sum_{l} x_{lc} \leqslant 1 \quad \forall c \tag{3}$$

$$\sum_{c} x_{lc} \leqslant 1 \quad \forall l \tag{4}$$

$$\sum_{l} \sum_{c} w_c x_{lc} \leqslant Q \tag{5}$$

$$\sum_{c \in T} x_{ijkc} = 0 \quad \forall i \in E, j, k \tag{6}$$

$$\sum_{c \in F} x_{ijkc} = 0 \quad \forall i \in O, j, k \tag{7}$$

$$\sum_{c \in T} x_{i+1jkc} + \sum_{c \in F} x_{ijkc} \leqslant 1 \quad \forall i \in E, j, k \tag{8}$$

$$\sum_{c \in T} x_{i-1jkc} + \sum_{c \in F} x_{ijkc} \leqslant 1 \quad \forall i \in E, j, k \tag{9}$$

$$\sum_{c \in T} x_{i+1jk+1c} + \sum_{c \in F} x_{ijkc} \leqslant 1 \quad \forall i \in E, j, k = 1, \ldots, |K| - 1 \tag{10}$$

$$\sum_{c \in T} x_{i-1jk+1c} + \sum_{c \in F} x_{ijkc} \leqslant 1 \quad \forall i \in E, j, k = 1, \ldots, |K| - 1 \tag{11}$$

$$\sum_{c \in T} w_c x_{ijkc} + \sum_{c \in T} w_c x_{ijk+1c} + \sum_{c \in T} w_c x_{ijk+2c} \leqslant MT \quad \forall i, j, k = 1, \ldots, |K| - 2 \tag{12}$$

$$\sum_{c \in F} w_c x_{ijkc} + \sum_{c \in F} w_c x_{ijk+1c} + \sum_{c \in F} w_c x_{ijk+2c} \leqslant MF \quad \forall i, j, k = 1, \ldots, |K| - 2 \tag{13}$$

$$\sum_{\substack{c,e \in C: \\ w_c \neq w_e}} (w_c x_{ijkc} - w_e x_{ijk+1e}) \geqslant 0 \quad \forall i, j, k = 1, \ldots, |K| - 1 \tag{14}$$

$$\sum_{\substack{c,e \in C: \\ d_c \neq d_e}} (d_c x_{ijkc} - d_e x_{ijk+1e}) \geqslant 0 \quad \forall i, j, k = 1, \ldots, |K| - 1 \tag{15}$$

$$-Q_2 \leqslant \sum_{i \in A, j, k} \sum_{c} w_c x_{ijkc} - \sum_{i \in P, j, k} \sum_{c} w_c x_{ijkc} \leqslant Q_2 \tag{16}$$

$$-Q_1 \leqslant \sum_{i, j \in L, k} \sum_{c} w_c x_{ijkc} - \sum_{i, j \in R, k} \sum_{c} w_c x_{ijkc} \leqslant Q_1 \tag{17}$$

$$x_{lc} \in \{0, 1\} \quad \forall l, c \tag{18}$$

(1) is the objective function that minimises the total stowage time L, expressed in terms of the sum of time t_{lc} required for loading a container c, $\forall c \in C$, in location l, $\forall l \in S$. Note that t_{lc} is given by

the time for handling container c by a yard transtainer and its positioning on board and depends, as we will see later, on the row and tier address in the ship.

Constraint (2) defines the number of locations to select for stowing the given containers. Relations (3) and (4) are the well known assignment constraints forcing each container to be stowed only in one ship location and each location to have at most one container.

The capacity constraint (5) establishes that the total weight of all containers cannot exceed the maximum weight capacity Q of the containership.

Relations (6)–(11) are the size constraints, as they have been described in Section 2. In particular, (6) and (7) force, respectively, $40'$ containers to be stowed in even bays and $20'$ container to be stowed in odd bays, while (8) and (9) make unfeasible the stowage of $20'$ containers in those odd bays that are contiguous to even locations already chosen for stowing $40'$ containers, and inversely; (10) and (11) prevent $20'$ containers being positioned over $40'$ ones.

Weight constraints (12) and (13) say that a stack of at most three containers of either $20'$ or $40'$ cannot exceed values MT and MF, respectively, that usually correspond to 45 and 66 tons; note that such constraints verify the corresponding tolerance value in all occupied tiers in the same row and bay for any possible stack of three containers, as it is required by the weight constraints described in Section 2. Constraints (14) force heavier containers not to be put over lighter ones. It is worth noting that constraints (14) also avoid the stowing of both $20'$ and $40'$ containers over empty locations.

The destination constraints (15) avoid positioning containers that have to be unloaded first below those containers that have a later destination port.

Constraints (16) and (17) are the horizontal and cross equilibrium conditions, stating that the difference in weight between the anterior and the posterior bays and between the left and right side must be at most Q_2 and Q_1 tons respectively. Note that in this model the condition related to the vertical equilibrium given in Section 2 can be dropped since it becomes redundant due to the weight constraint given in (14).

Finally, in (18) the binary decision variables of the problem are defined.

Note that in the formulation of the problem we assume that the ship starts its journey in the port for which we are studying the problem and successively visits a given number of other ports where only unloading operations are allowed. We can justify this assumption by remembering that we are involved with the stowage planning problem of a terminal that is not really affected by what happens in the next ports, and hence we do not interact with the ship coordinator. Moreover, we assume that the number of containers to load on board is not greater than the number of available locations; this means that we are not concerned with the problem of selecting some containers to be loaded among all and that the capacity constraint (5) is only related to the maximum weight available for all containers. In the next Section we will verify condition (5) in order to prevent unfeasibilities due to constraint (2).

4. The solution approach

The proposed solution approach for MBPP consists of three main phases. The first phase, as described in Section 4.1, is the instance's preprocessing aimed at tightening the constraints of the above model and reducing the feasible region of each container $c \in C$. Note that the above LP model is aimed at defining the stowage plan only for standard containers, while in the preprocessing step of our procedure it is possible to assign more then one location at out of gauge containers (like hazardous or reefer containers).

The second phase, described in Section 4.2, is the prestowage procedure, where we consider set S of available locations split into different partitions with respect to their bay address. This is done in such a way as to make easier the stowage of containers according to their destination. Finally, we solve the *MBPP Model* given in Section 3; a modified version taking into proper account the previous preprocessing and prestowing procedures.

4.1. Preprocessing

First of all, we sort the data related to the available ship locations according to their bay address, in such a way as to look at the location of the containers with respect to indices i, j and k, respectively. The final solution hence gives a picture of the loaded ship as it is ready for its journey, that is the master bay plan giving for each container its position on board.

Then, in this phase we remove from set S all potential locations that a priori are not to be considered for stowing container c, $\forall c \in C$. That is we do not have variable x_{lc} if location l cannot stow container c (i.e. it is preset at zero for that value of c). Moreover, we consider the specifications for both reefer and hazardous containers by setting to one the corresponding variables before solving the model. For instance, setting $x_{01,02,02,004} = 1$ assigns to bay 01, row 02, tier 02 the container with code number 004 that is a container for dangerous loads. Consequently, according to the size constraints (see Section 2) we can a priori remove all variables related to both odd bays for 40′ containers and even bays for 20′ ones, since they would be forced to zero in any feasible solution (see constraints (6) and (7)).

In practice, in this phase, we initially assign set S to $\overline{S}$, where $\overline{S}$ denotes the updated set of available locations (variables) for stowing, and remove from $\overline{S}$ location l_{ijkc} if the following conditions (19) and (20) are satisfied.

$$\text{if } (F \neq \emptyset) \text{ then for } (i \in O; c \in F; \forall j,k)\ \overline{S} := \overline{S} \setminus \{l_{ijk}\}; \tag{19}$$

$$\text{if } (T \neq \emptyset) \text{ then for } (i \in E; c \in T; \forall j,k)\ \overline{S} := \overline{S} \setminus \{l_{ijk}\}. \tag{20}$$

In particular, condition (19) makes unfeasible the stowage of 40′ containers in odd bays. Analogously, condition (20) forces 20′ containers to be stowed only in odd bays.

Finally, since we assume in this work that all containers to be stowed are ready in the storage area at the quay, we have to verify whether the total weight $W = \sum_c w_c$ of the containers does not exceed the maximum capacity Q of the ship. If this is not the case, then capacity constraint (5) is a redundant constraint and hence removed and the assignment constraints (3) are put as equality constraints, since all containers have to be stowed; otherwise, constraint (2) is put as an inequality condition defining the upper bound for the number of locations to be chosen.

In practice in this phase we make redundant constraints (6), (7) and either (2) or (5) and tight the constraints of *Model MBPP*. Note that, in particular, the relations (14) make redundant constraints (10) and (11) since it implicitly includes them, and constraint (2) together with constraints (8) and (9) make redundant the assignment conditions (4).

4.2. The prestowing procedure

Apart from the basic combinatorial constraints (3)–(5), that are the assignment and knapsack constraints, the destination constraints (15), that are relevant for the master bay plan problem, are active combinatorial constraints and strongly affect the computational time (see Sections 5 and 6). Note that these last constraints force containers stowed in the highest tiers to be unloaded first and hence, from an operational point of view, they are very important. In fact, without considering the unloading port some containers loaded last would necessarily have to be moved for enabling the unloading operations of some others. As has been mentioned in Section 1, "empty moves" are very unproductive for any maritime company because they increase the berthing time at port, thus affecting the cost of the whole trip of the containership.

In order to avoid unproductivity due to a possible relaxation of the destination constraints, we perform a heuristic prestowing procedure. In particular, we force containers to be located only in some bays that are established according to their destination. In this way, the search for the location of each container is limited to a portion of the ship and, therefore, the space of the solution of the problem is reduced and thus its complexity.

Practically, this procedure can be view as the execution of certain stowing instructions given by the ship coordinator and by the planning office that determine the loading plan. These instructions define a prestowage and based on them the final stowage plan is determined.

Presently, we briefly report the main steps of our prestowing procedure, that is described in more detail in Ambrosino et al., 2002.

Step 1: sorting C. We split set C of all containers to be stowed in the ship into p subsets C_h, $h = 1, \dots, p$, where p is the number of different ports visited by the ship, such that $c \in C_h$ if and only if $d_c = h$, $\forall h = 1, \dots, p$. This means that containers are grouped together according to their destination, so that $\bigcup_{h=1,\dots,p} C_h = C$ and $C_g \cap C_h = \emptyset$, $\forall h \neq g$, $h, g = 1, \dots, p$. We then sort C in such a way that $C_h < C_g$ if and only if $h < g$, $h \neq g$, $h, g = 1, \dots, p$. Elements belonging to C_1 are hence containers to be unloaded first, elements belonging to C_2 are containers having as destination the second port visited by the ship, and so on.

Step 2: partitioning I. We consider set I of all available bays of the ship as b different partitions, where $b = |I|$, and associate with each of them subset C_h, $h = 1, \dots, p$, depending on the value of p. In this way we make unfeasible the stowage of containers out of their predefined bays, thus reducing the decision space of each variable of the problem.

In particular, our idea consists of first assigning the central bays of the ship, namely bays $b/2$ and $b/2 + 1$ to C_1, that is to containers that have to be unloaded first; this means that all tiers and rows of bays $b/2$ and $b/2 + 1$ are populated by containers belonging to C_1. Then, if $p = 2$, we assign alternatively their contiguous bays to C_2 and C_1. Otherwise, that is if $p \geqslant 3$, starting from bay $b/2$, we assign bay $b/2 - 1$ to C_2 and bay $b/2 + 2$ to C_3, and continue to assign alternating bays to different subsets depending on their cardinality.

An example of different assignments of set I of bays is reported in Fig. 3, where only even bays are conventionally considered. Fig. 3 refers to those few instances in which either 2 or 3 different ports have to be visited by a containership.

Note that this bay assignment procedure follows the basic criterion by which containers to be unloaded first are located in the central bays in order to avoid, as far as possible, imbalances in the bow–stern weight distribution. In the bays close to the centre on one side are located containers

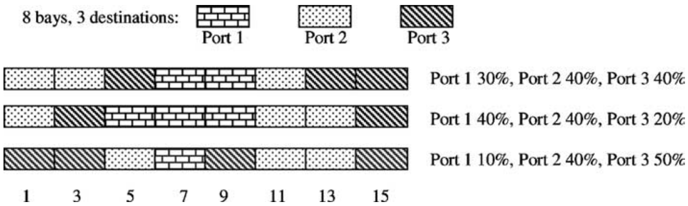

Fig. 3. Example of bay assignments in the prestowing procedure.

having the second port as a destination and on the other containers to the third port. This avoids imbalances after the first and later containers unloading. This choice also enables easier and faster container loading in the ports visited after the original loading port.

At the end of this step we then have set I_{C_h}, $h = 1, \dots, p$, that identifies all bays where containers that have as destination port h must be stowed. In order to prevent unfeasibility, we keep free some locations in the highest tiers, in such a way that only about 80% of the total locations are assigned a priori to subsets C_h, $h = 1, \dots, p$. In practice, in our procedure we do not consider the last tier in each row and bay. This choice is justified by the fact that the loading time, for instance by a yard transtainer, in the upper tiers is as low as possible and hence in the case where an empty move is required it does not affect the overall loading time value L given by (1) too much.

Step 3: reducing S. Analogous to what has been done in the preprocessing phase (see Section 4.1) we remove from $\overline{S}$ those locations having as bay address (i.e. index i) values that do not correspond to the bay-container assignment defined in Step 2. Consequently, all variables x_{ijkc} such that $i \notin I_{C_h}$, $\forall c \in C_h$, are not generated.

The whole prestowing procedure is here below synthetized in a C-like formulation.

Procedure (Prestowing)
begin
  /* *group containers according to their destination* */
  split C into p subsets such that $c \in C_h$ iff $d_c = h\ \forall h = 1, \dots, p$;
  sort C in an increasing order such that $C_h < C_g$ iff $h < g\ h \neq g,\ h,g = 1, \dots, p$;
  /* *group the bays according to the containers destination* */
  assign bays $b/2$ and $b/2 + 1$ to C_1;
  if ($p >= 3$) *then* assign bay ($b/2 - 1$) to C_2 and bay ($b/2 + 2$) to C_3;
    else assign bays ($b/2 - 1$ and $b/2 + 2$) to C_2;
  for ($i = 1$ to $b/2 - 1$) and ($i = b/2 + 3$ to b)
    for ($h = p$ down to 2)
      assign alternatively bay i to C_h;
  /* *remove locations of unfeasible bay-destination assignments* */
  for ($h = 1$ to p)
    for ($i \notin I_{C_h}$, $j \in J$, $k \in K \setminus \{\text{last tier}\}$, $c \in C_h$) *do*
      $\overline{S} := \overline{S} \setminus \{l_{ijk}\}$
end.

4.3. Solution of the problem

Finally, we solve *MBPP Model* according to the modifications performed in the previous phases, that is with the relaxation of constraints (4)–(7), (10), (11) and (15), considering set $\overline{S}$ instead of the original set S of locations with subsets C_h, $h = 1, \ldots, p$ and their corresponding feasible bay assignment I_{C_h}.

5. Performance evaluation

5.1. A simple case study

To give an idea of the type of problem we are involved with, let us first present a simple case study concerning the stowage plan of a 64 TEU containership, where 15 standard containers, split between 20′ and 40′ with weight ranging from 10 to 25 tons, have to be loaded for having 1 or 2 ports to visit (see Table 1). The ship consists of four odd bays, namely 1, 3, 5 and 7, that yield two even bays, namely 2 and 6, four rows, namely R03, R01, R02 and R04, and four tiers, denoted by 02 and 04 in the hold and 82 and 84 in the upper deck. The ship has a maximum capacity of 250 tons and the maximum cross and horizontal weight tolerance is fixed to 20 and 40 tons, respectively (i.e. $Q_1 = 20$ and $Q_2 = 40$). The loading times with respect to each tier and row are reported in Table 2, where we can see that they are independent from the bay and increase when the tier is lower and the row is further from the yard side.

The formulation of the problem according to *Model MBPP* results in 1440 x_{cl} variables and 1154 constraints. The optimal solution, corresponding to the objective function value (1), is $L^* = 33.8$ (that is 33 min and 48 s). While using the heuristic approach proposed in Section 4 the minimum loading time is $L = 34$, corresponding to an optimality gap of 0.59%. The relative

Table 1
Data of the simple case study

Container	Size	Destination	Weight
1	20′	1	10
2	20′	1	10
3	20′	1	10
4	20′	2	10
5	20′	2	10
6	40′	1	20
7	40′	2	20
8	40′	2	10
9	20′	1	15
10	20′	1	15
11	20′	2	10
12	20′	2	10
13	40′	1	25
14	40′	1	25
15	40′	2	20

Table 2
Loading times as a function of row and tier addresses

	Tier 02	Tier 04	Tier 82	Tier 84
Row 04	2′ 36″	2′ 30″	2′ 24″	2′ 18″
Row 02	2′ 30″	2′ 24″	2′ 18″	2′ 12″
Row 01	2′ 24″	2′ 18″	2′ 12″	2′ 06″
Row 03	2′ 18″	2′ 12″	2′ 06″	2′

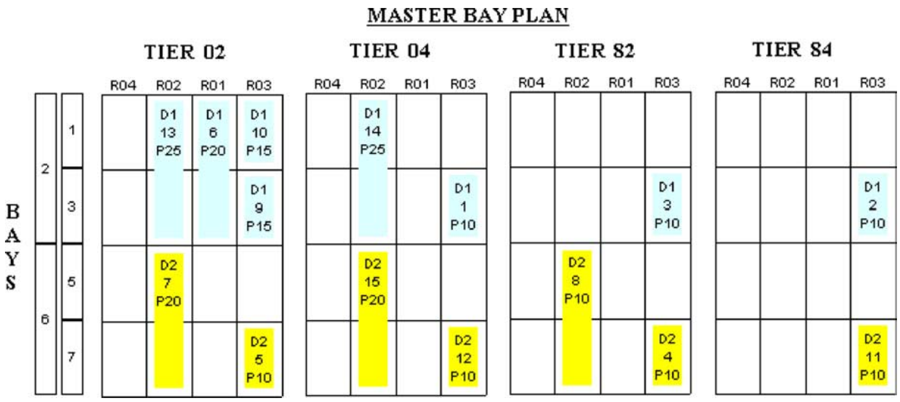

Fig. 4. Stowage plan of the simple case.

computational times are CPU* = 7′ 54″ and CPU = 11″, respectively, obtained on a PC Pentium II, using MPL (Maximal Software, 2000) and Cplex, together with the heuristic procedures written in the C++ language. Our solution is shown in Fig. 4, which depicts the corresponding master bay plan.

5.2. Computational experiments

As a comparison and evaluation of the prestowing procedure we have generated 12 classes of medium sized scenarios, each one with either two or three destinations, with containers having different destination, weight and size. Such scenarios have been solved by first using *Model MBPP*, then relaxing the assignment constraints (4) and the destination ones (15), and finally executing the procedure described in Section 4. All computational experiments have been performed on the same platform as before. The results are shown in Table 3, where each value is the average of seven entries. The headings are as follows: columns *Destination*, *Weight* and *Size* give for each class of scenario the specification of the set of containers to be loaded on the ship; L^* is the optimal loading time (in minutes), (i.e. the solution of *Model MBPP*), L° is the objective function value (1) of the relaxed model without constraints (15), that is our lower bound, and L is the same value obtained by applying our proposed approach to the problem resolution. These three objective function values are also graphically reported in Fig. 5. Notice the good behaviour of our approach, expecially when containers are heterogeneous and belong to all classes of weight,

Table 3
Computational results of the different resolution approaches with two and three destinations

Case	Destination			Weight			Size		Objective function (L)			Δ_{opt}%		Computational time (CPU)			%Unfeas
	d_1	d_2	d_3	lw	mw	hw	20′	40′	L^*	L°	L	$\frac{\lvert L^\circ - L^* \rvert}{L^\circ}$	$\frac{L-L^*}{L}$	CPU*	CPU°	CPU	
1—2 dest	5	5	/	/	10	/	10	/	22.5	22.5	22.6	0.0	0.4	22.04″	12.34″	4.08″	10
1—3 dest	3	4	3	/	10	/	10	/	22.5	22.5	22.6	0.0	0.4	38.33″	13.26″	4.32″	20
2—2 dest	5	5	/	/	10	/	8	2	22.5	22.5	22.7	0.0	0.9	9.11″	3.56″	2.25″	10
2—3 dest	3	4	3	/	10	/	8	2	22.5	22.5	22.8	0.0	1.3	9.46″	4.06″	2.42″	20
3—2 dest	5	5	/	/	10	/	5	5	22.6	22.6	22.8	0.0	0.9	112.23″	55.18″	6.02″	20
3—3 dest	3	4	3	/	10	/	5	5	22.6	22.6	22.9	0.0	1.3	146.15″	54.17″	7.16″	20
4—2 dest	5	5	/	/	10	/	2	8	22.5	22.5	22.8	0.0	1.3	8.45″	4.11″	2.45″	0
4—3 dest	5	3	2	/	10	/	2	8	22.5	22.5	22.8	0.0	1.3	8.52″	4.52″	2.53″	10
5—2 dest	5	5	/	4	4	2	10	/	22.5	22.2	22.5	1.4	0.0	45.23″	24.35″	5.15″	10
5—3 dest	3	4	3	4	4	2	10	/	22.6	22.2	22.7	1.8	0.4	88.52″	24.52″	6.01″	20
6—2 dest	5	5	/	4	4	2	8	2	22.6	22.6	22.8	0.0	0.9	55.08″	22.58″	4.25″	20
6—3 dest	3	4	3	4	4	2	8	2	22.8	22.6	23.0	0.9	0.9	74.16″	22.52″	5.14″	20
7—2 dest	5	5	/	4	4	2	5	5	22.8	22.6	23.2	0.9	1.7	69.19″	31.51″	4.58″	20
7—3 dest	3	4	3	4	4	2	5	5	22.9	22.6	23.0	1.3	0.4	98.35″	32.32″	5.49″	30
8—2 dest	5	5	/	4	4	2	2	8	22.7	22.5	22.8	0.9	0.4	52.48″	26.47″	3.21″	20
8—3 dest	5	3	2	4	4	2	2	8	22.7	22.5	22.8	0.9	0.4	74.08″	28.02″	3.55″	30
9—2 dest	5	5	/	2	4	4	10	/	23.0	22.8	23.0	0.9	0.0	12.48″	7.01″	2.58″	10
9—3 dest	3	4	3	2	4	4	10	/	23.0	22.8	23.2	0.9	0.9	15.03″	7.12″	3.22″	20
10—2 dest	5	5	/	2	4	4	8	2	22.8	22.8	23.2	0.0	1.7	23.16″	16.34″	4.12″	20
10—3 dest	3	4	3	2	4	4	8	2	23.0	22.8	23.2	0.9	0.9	31.05″	17.05″	5.29″	30
11—2 dest	5	5	/	2	4	4	5	5	23.0	22.6	23.0	1.8	0.0	31.52″	11.53″	3.42″	30
11—3 dest	3	4	3	2	4	4	5	5	23.2	22.6	23.4	2.7	0.9	34.53″	10.58″	4.03″	40
12—2 dest	5	5	/	2	4	4	2	8	22.9	22.5	22.9	1.8	0.0	88.02″	34.42″	5.50″	30
12—3 dest	5	3	2	2	4	4	2	8	22.9	22.5	22.9	1.8	0.0	95.38″	34.16″	5.58″	30
Average	4	4	1	2	6	2	6	4	22.73	22.56	22.90	0.78	0.73	51.77″	20.90″	4.01″	20

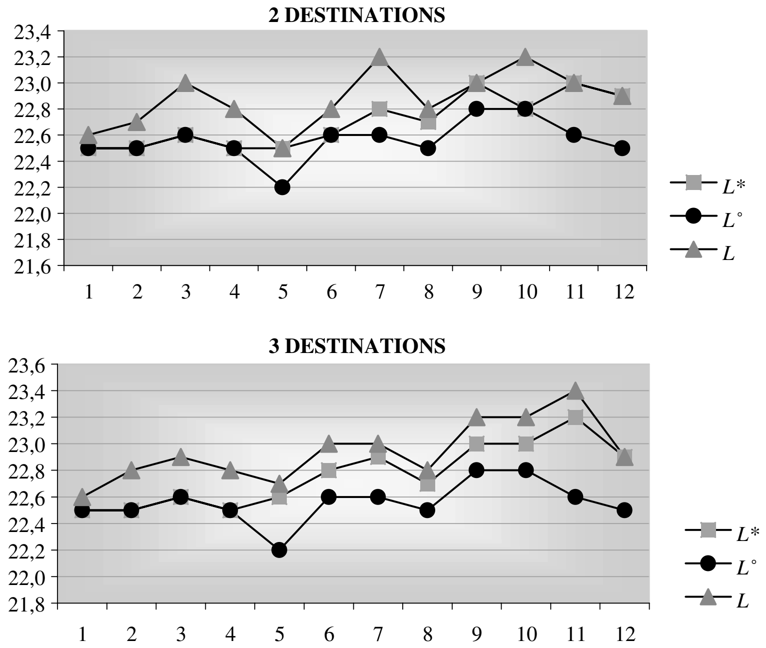

Fig. 5. Trend of the objective function's value for the 12 considered classes of scenario.

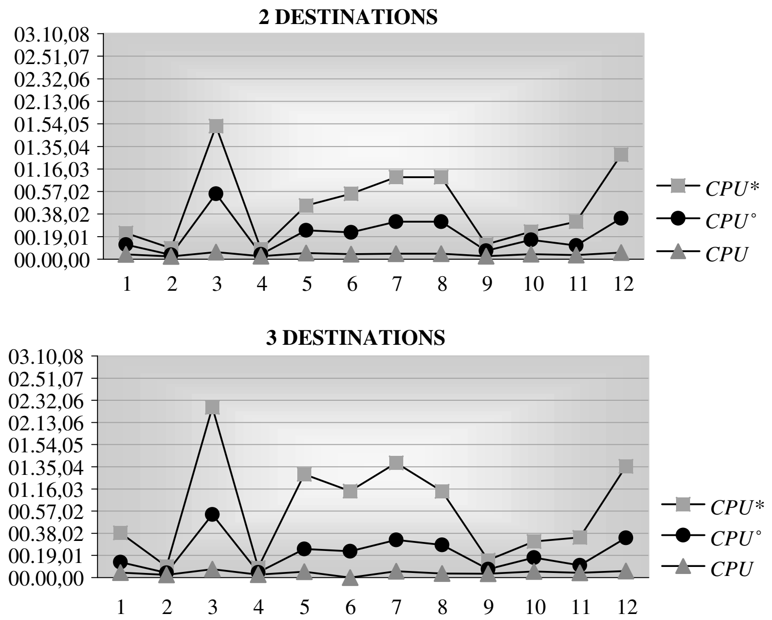

Fig. 6. Trend of the computational time for the 12 considered classes of scenario.

and how close is L to L^* even in the worst case, that is 1.7% for cases 7 and 10. The good performance of the heuristic approach can be observed in column Δ_{opt} that reports the optimality gap as a percentage difference, between the optimal solution and both the relaxed and the heuristic one, given respectively by the ratios $\frac{L^*-L^\circ}{L^\circ}\%$ and $\frac{L-L^*}{L}\%$.

CPU*, CPU$^\circ$ and CPU are the computational times, in seconds, corresponding to the above solutions. The computational times in the three cases are also graphically reported in Fig. 6, which outlines the impressive reduction of the computational time when our approach is used.

As a further element of the analysis, a possible violation of destination constraints by the optimal solution in the relaxed model has been detected. Such unfeasibilities, whose percentage is reported in column %*Unfeas*, have been removed by changing the relative position of pairs of containers, first in all tiers in each bay for all bays, and successively in each tier among different bays. This search approach implies $m(m-1)/2$ comparisons of values d_c, $\forall c \in C$. The next research direction of the authors will be to investigate the applicability and the performance of multiexchange local search techniques, as proposed in Ambrosino (2001), for removing unfeasibilities and solving MBPP by a more ambitious research approach.

6. The operational scenario: a maritime terminal in genoa

In this section we present a case study which motivated our analysis of MBPP. In particular, we consider the Chiwaua containership, that is a "client" of the maritime terminal in Genoa (Italy). It is a 198 TEU containership, with 11 bays, four rows and five tiers (three in the hold and two in the upper deck, respectively). All structural and operational information about the Chiwaua ship are available to interested readers. We test our approach deducing the master bay plan by referring to 13 cases, reported in Table 4. As you can see, such cases differ from each other by the number of containers to load, ranging from 100 to 158, their size and weight, the number of ports to be visited, that is either 2 or 3, and the number of TEUs to load on board, ranging from 138 to 188.

Table 4
Data of the Chiwaua case under consideration

Case	Container									TEUs	Full (%)
	Size			Weight			Destination				
	Tot	20′	40′	lw	mw	hw	d_1	d_2	d_3		
1	100	62	38	45	30	25	47	53	0	138	73.4
2	120	75	45	50	44	26	55	65	0	165	87.7
3	130	90	40	56	46	28	55	75	0	170	90.0
4	130	85	45	58	45	27	62	68	0	175	93.0
5	130	75	55	56	46	28	60	70	0	185	98.4
6	135	84	51	55	50	30	60	75	0	186	98.9
7	140	100	40	65	47	28	61	79	0	180	95.7
8	140	95	45	60	50	30	65	75	0	185	98.4
9	140	95	45	60	50	30	50	40	50	185	98.4
10	148	108	40	62	53	33	69	77	0	188	100
11	148	108	40	62	53	33	50	40	48	188	100
12	150	120	30	62	55	33	65	85	0	180	95.7
13	158	128	30	62	63	33	75	83	0	188	100

Column *Full* gives the ship occupation level (in percentage terms) when all containers are loaded. A 100% occupation level is allocated when 188 TEUs are loaded since, conventionally, 10 TEU locations are operationally always kept free for security and possible emergency reasons. From such as index, it is possible to derive the number of remaining free locations, considering that each container of 40 feet needs two cells and hence two TEUs.

The computational results related to the cases described in Table 4 are reported in Table 5. As before, Table 5 gives the loading time L^* (in minutes) of the optimal value (1) obtained by solving *Model MBPP*, the lower bound L° and the objective function value L obtained by applying the approach presented in Section 4, as well as their relative optimality gaps, as described in Section 5.2. L is very close to L^*, and differs from it by an average of 0.22%.

It is also interesting to analyse the loading time as a function of the percentage of occupation of the ship (see column *Full* in Table 4). The difference in loading times between two contiguous locations is generally of the order of 6 s (see Table 2) and increases when we move from the quay side towards the bottom of the ship, since the locations become more difficult to reach (see Fig. 1). This means that the optimal loading time, is in some sense, not too much affected by the chosen approach when the ship occupation level is about 100%. This is because the problem is more constrained and the choice with respect to the loading time minimization is very limited. In fact, when the ship is full the value of the objective function is just the sum of the times, given as input data, required for handling the containers by the available yard transtainers from the quay to their location on the ship. The locations chosen for stowing 40′ containers play a relevant role in the minimization criterion. Consequently, we try as far as possible to locate 40′ containers in lower tiers on the seaside for reducing the handling operations in the most time expensive locations for putting two 20′ containers.

In Table 5 columns CPU*, CPU$^\circ$ and CPU report the computational time, in minutes, of the corresponding solution. The impact of the destination constraints (15) on it can easily be seen. For

Table 5
Computational results for the Chiwaua case

Case	L^*	L°	L	$\frac{\lvert L^\circ - L^* \rvert}{L^\circ}$ (%)	$\frac{L-L^*}{L}$ (%)	CPU*	CPU$^\circ$	CPU	Handled
1	230.9	229.4	231.4	0.65	0.22	16′ 49″	12′ 16″	2′ 51″	25.97
2	276.4	274.5	278.7	0.69	0.83	19′ 10″	17′ 10″	8′ 17″	26.04
3	301.6	300.4	302.1	0.40	0.17	29′ 24″	22′ 31″	10′ 19″	26.07
4	300.8	300.1	301.0	0.23	0.07	30′ 48″	28′ 41″	2′ 56″	26.10
5	300.8	299.8	302.1	0.33	0.62	33′ 44″	20′ 35″	11′ 01″	26.20
6	312.0	311.1	312.0	0.29	0.00	29′ 30″	25′ 31″	5′ 31″	25.96
7	323.3	323.1	324.4	0.06	0.34	46′ 53″	28′ 58″	10′ 05″	26.00
8	324.0	324.0	324.9	0.00	0.28	26′ 21″	22′ 07″	15′ 23″	25.92
9	324.5	324.0	324.6	0.15	0.03	1 h 43′ 02″	22′ 08″	14′ 32″	25.88
10	341.9	340.5	342.4	0.41	0.15	37′ 43″	35′ 13″	13′ 19″	25.97
11	342.3	340.5	342.3	0.53	0.00	1 h 52′ 05″	35′ 21″	13′ 10″	25.94
12	348.1	347.6	348.2	0.14	0.03	48′ 01″	46′ 25″	14′ 16″	25.85
13	365.8	365.3	366.3	0.14	0.14	45′ 46″	40′ 15″	33′ 53″	25.91
Average	291.7	290.8	292.2	0.31	0.22	44′ 34″	27′ 29″	11′ 58″	25.99

example, cases 9 and 11, both with three different ports to be visited by the ship it reduces respectively from 103 to 22 min in the relaxed model, and from 112 to 35 min on the same platform as in Section 5. Moreover, see how the proposed heuristic approach further improves the computational time in all instances, even if CPU is just the time required by Cplex and does not include the time of our preprocessing/prestowing procedures, which takes only few seconds in all cases. It is worth mentioning that the planning office of the terminal that has provided the data for the Chiwuaua containership takes about from 60 to 90 min for manually compiling the corresponding master bay plans.

Finally, with reference to the objective function value (1) it is possible to evaluate a very important terminal index, reported in column *Handled* in Table 5, that is the number of handling operations per hour. Note that, on average, by applying our approach we have almost 26 container movements/hour, while the same index in the present operational scenario at the maritime terminal is about 24. This means that our approach not only enables the obtaining of good solutions in a very short computational time but also guarantee better terminal performance indices than those that are provided today.

As a further consideration related to the Chiwuaua containership, it is important to mention that the implementation of *Model MBP* requires 28,500 variables for case 1 up to 45,030 for case 13, and from 14,015 to 18,231 constraints for the same cases.

7. Concluding remarks

In this paper the problem of stowing containers into a containership has been faced by evaluating an exact 0–1 Linear Programming model, that is not practically useful for large cases. This has been modified by an approach proposed by the authors, consisting of heuristic preprocessing and prestowing procedures that also allow the relaxation of some constraints of the exact model. The proposed approach exhibits very good performance in terms of both solution precision and computational time. Moreover, in the performance evaluation with real size cases, it also guarantees another very important maritime performance index, in term of handling operations/hour.

We think that the proposed approach is very valuable and that one of the future directions of this study should be its application to evaluating the impact of the ship system requirements on the whole organization of the yard.

Acknowledgements

The authors wish to thank the anonymous referees for their valuable comments that help in improving the quality of the paper.

References

Ambrosino, D., 2001. A local search algorithm for a complex distribution network design problem. Preprints of the IV Triennal Symposium on Transportation Analysis, Ponta Delgada, Azores Island, pp. 479–484.

Ambrosino, D., Sciomachen, A., 1998. A constraints satisfaction approach for master bay plans. In: Sciutto, G., Brebbia, C.A. (Eds.), Maritime Engineering and Ports. WIT Press, Boston, pp. 155–164.

Ambrosino, D., Sciomachen, A., Tanfani, E., 2002. Optimization containership stowage plans: linear model and heuristic procedures. In: Proceedings of 3nd International Congress on Maritime Technological Innovations and Research, Bilbao, pp. 7–16.

Atkins, W.H., 1991. Modern Marine Terminal Operations and Management. Boyle, Oakland.

Avriel, M., Penn, M., 1993. Exact and approximate solutions of the container ship stowage problem. Computers & Industrial Engineering 25, 271–274.

Avriel, M., Penn, M., Shpirer, N., 2000. Container ship stowage problem: complexity and connection to the colouring of circle graphs. Discrete Applied Mathematics 103, 271–279.

Bischoff, E.E., Mariott, M.D., 1990. A comparative evaluation of heuristics for container loading. EJOR 44, 267–276.

Bischoff, E.E., Ratcliff, M.S.W., 1995. Issues on the development of approaches to container loading. International Journal of Management Science 23 (4), 377–390.

Bortfeldt, A., Gehring, H., 2001. A hybrid genetic algorithm for the container loading problem. EJOR 131 (1), 143–161.

Chen, C.S., Lee, S.M., Shen, Q.S., 1995. An analytical model for the container loading problem. EJOR 80 (1), 68–76.

Crainic, G., Gendreau, M., Dejax, P., 1993. Dynamic and stochastic models for the allocation of empty containers. Operations Research 41, 102–126.

Davies, A.P., Bischoff, E.E., 1999. Weight distribution considerations in container loading. EJOR 114, 509–527.

Gehring, M., Bortfeldt, A., 1997. A genetic algorithm for solving container loading problem. International Transactions of Operational Research 4 (5/6), 401–418.

Gehring, M., Menscher, Meyer, M.A., 1990. Computer-based heuristic for packing pooled shipment containers. EJOR 44, 277–288.

Imai, A., Nishimura, E., Papadimitriu, S., Sasaki, K., 2002. The containership loading problem. International Journal of Maritime Economics 4, 126–148.

Kim, K.H., Park, Y.M., Ryu, K.-R., 2000. Deriving decisions rules to locate export containers in container yards. EJOR 124 (1), 89–101.

Maximal Software, Inc., 2000. MPL Modeling System, USA.

Preston, P., Kozan, E., 2001. An approach to determine storage locations of containers at seaport terminals. Computers and Operations Research 28 (10), 983–995.

Raidl, G.R., 1999. A weight-coded genetic algorithm for the multiple container packing problem. In: Proceedings of the 1999 ACM Symposium on Applied Computing, San Antonio, Texas, pp. 291–296.

Taleb-Ibraimi, Mounira, De Castilho, B., Daganzo, C.F., 1993. Storage space vs handling work in container terminals. Transportation Research B 27, 13–32.

Thomas, B.J., 1989. Management of Port Maintenance: A Review of Current Problems and Practices. H.M.S.O, London.

Wilson, I.D., Roach, P., 2000. Container stowage planning: a methodology for generating computerised solutions. Journal of the Operational Research Society 51 (11), 248–1255.