Knowledge-Based Systems 14 (2001) 137–145

Knowledge-Based SYSTEMS

www.elsevier.com/locate/knosys

Container stowage pre-planning: using search to generate solutions, a case study

I.D. Wilson*, P.A. Roach, J.A. Ware

Department of Mathematics, University of Glamorgan, Pontypridd, Mid Glamorgan CF37 1DL, UK

Accepted 2 February 2001

Abstract

Container-ships are vessels possessing an internal structure that facilitates the handling of containerised cargo. At each port along the vessel's journey, containers destined for those ports are unloaded and additional containers destined for subsequent ports are loaded. Determining a viable arrangement of containers that facilitates this process, in a cost-effective way, constitutes the deep-sea container-ship stowage problem. This paper outlines a computer system that generates good sub-optimal solutions to the stowage pre-planning problem. This is achieved through an intelligent analysis of the domain allowing the problem to be divided into sub-problems: a generalised placement strategy and a specialised placement procedure. This methodology progressively refines the arrangement of containers within the cargo-space of a container ship until each container is specifically allocated to a stowage location. Good, if not optimal, solutions for the problem are obtained in a reasonable processing time through the use of heuristics incorporated into objective functions for each stage.

Keywords: Planning; Transportation; Container

1. Introduction

Containerisation (the packing of cargo into large, dedicated boxes, of different dimensions, enabling multiple units of cargo to be handled simultaneously) has increasingly facilitated the transportation of cargo since the 1970s. In order to increase the benefits of economy of scale, the size of *container ships* has increased.

This increase in capacity has seen movement from relatively small ships with 350 Twenty Foot Equivalent Units (TEUs) to ships with capacities of more than 4500 TEUs [1]. Increasing standardisation of containers has permitted the introduction of *inter-modal* transportation systems. That is, the standard frame and dimensions of containers allows containerised cargo to be transported by rail, truck or sea.

Container ships travel on 'round-robin' routes where at each port of destination (POD) containers may be unloaded and additional containers destined for subsequent ports may be loaded. Determining a viable arrangement of containers that facilitates this process, in a cost-effective way, makes up the container stowage problem. Human stowage planners determine a stowage arrangement for a container ship. These planners work under demanding time constraints, and are limited in the number of arrangements that they can consider.

Modern container ships can require thousands of container *movements*(the loading, unloading or re-positioning of each container) at each POD to complete the discharge and load process. (Fig. 1 shows such a ship from above and in transverse, longitudinal (along the length of the ship) and latitudinal (along the width of the ship) section; annotations mark positions of 'hatches' and 'bays' that are groupings of physical locations for container stowage).

Determining the arrangement of containers is an error-prone process relying on the intuitive skills of human planners. Planners must ascertain the placement of containers so that all *constraints* (restrictions placed upon where and how containers can be stowed) are satisfied and *material handling costs* (the costs associated with loading, unloading and transporting cargo) are minimised. The most important aspect of this optimisation process is the *re-handle*. A re-handle is a container movement made in order to permit access to another, or to improve the overall stowage arrangement, and is considered a product of poor planning.

The purpose of this paper is to highlight the complexity of the deep-sea containers-ship stowage problem, and to demonstrate how suitable objective functions can be constructed to facilitate its solution.

* Corresponding author. Tel.: +44-1443-480480; fax: +44-1443-482711.

E-mail address: idwilson@glam.ac.uk (I.D. Wilson).

0950-7051/01/$ - see front matter
PII: S0950-7051(01)00090-9

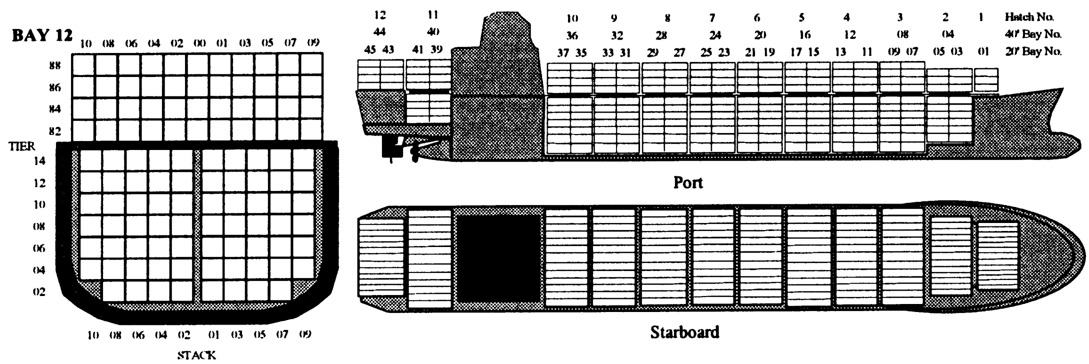

Fig. 1. Stowage arrangement for a container ship.

2. Problem complexity

The container stowage problem is a combinatorial problem the size of which depends upon ship capacity and the container supply and demand at each POD. Combinatorial optimisation is made more complicated by the need to consider stowage across a number of ports. A decision made at one port will have consequences at subsequent ports. Planners will consider a fixed number of subsequent ports when planning stowage. Hence a full definition of the deep-sea container-ship stowage problem is the determination of a stowage arrangement for a container ship, on leaving a port, so that no ship stability and stress constraints are violated, and efficiency is optimised.

Even for the smallest cases, container stowage planning is a large-scale problem. The following points outline the main constraints and guidelines, common to most operators, which must be considered by planners during the stowage planning process for an individual port [2].

The number of times a container must be re-handled before discharge is to be minimised. Reducing the number of re-handles results in large cost savings, though constraints will usually make it impossible to reduce this to zero. However, a stowage plan that minimises re-handles may itself be inefficient if the number of moves made by a crane and the distance travelled by it is excessive. Whereas it is sensible to group together cargo with the same destination in the same bay, a good disposition of this cargo between bays will permit multiple cranes to work. An optimum separation of four bays between cranes is required to facilitate simultaneous operation. This parallelisation of the loading and unloading process will permit a faster turnover of container movements to take place. Poor block stowage of cargo intended for the same destination may result in an excessive number of hatch-lid movements during unloading. Therefore, stowage should be planned so that hatch-lid movement is efficient.

Gradation in weight should be observed — that is, heavier containers should generally be placed at the bottom, and maximum allowable stack weights should not be exceeded. Stack height restrictions are to be observed and special consideration is to be given where crane height may be less than normal stack height. Vessels normally have $40'$ units placed on top of $20'$ units. Where $20'$ units are of a different height, $6''$ filler pieces can be used to bring the containers up to the required height. Stacks may not be completely filled due to stack weight limits, so stowage planning should ensure the maximum use of TEU and hence minimise the amount of lost cargo spaces. Ideally, only one discharge port's cargo should be stored under a single hatch (e.g. Hamburg). If this is not possible then the space should be taken up by cargo for another port with the furthest distance to travel (e.g. Hong Kong). 'Out of gauge' containers (of non-standard dimensions) should be placed at the top of container stacks, as this will minimise interference with adjacent slots. Empty and open top containers should usually be placed on top of stacks.

Cargo should only be placed in appropriate areas of the ship, although this is not always possible. For example, some cargo can only be placed in areas specifically allocated for its use. Each of the two types of *Reefer* unit (refrigerated container either independently powered or by the ship via a dedicated power outlet) should be stowed according to the appropriate rules and stowage requirements. On vessels that support this type of container, care must be taken to segregate the reefer commodities, so that tainting does not occur. Priority is given to placing reefers in designated reefer slots. Where possible, $40'$ reefer containers should be placed in stowage slots where only one reefer slot is used, rather than occupying two $20'$ reefer slots. Reefers should be stored away from locations that give off radiant heat, such as the Engine Room and Fuel Tanks. Empty reefers should occupy standard locations, i.e. not locations designated for reefer storage. *Fantainers* (containers that are ventilated by an internal fan) must be stowed near to reefer power points. Wet hides and wet salted hides tend to leak and give off a pungent odour [2]. Hides can only be stowed within cells that have been specially treated to receive them. Additionally, hides must always be at least two cells horizontally away from reefers or open topped containers, three bays away from crew accommodation, and are not allowed above or next to foodstuffs. Sometimes, so that vessel utilisation is maximised, containers may be stored in areas that are difficult to access at certain destinations (e.g. the berth at which the ship docks may not have cranes that can access an extreme part of a vessel).

A minimum distance must separate combinations of containers with hazardous cargo from other containers containing conflicting hazardous cargo. Stowage is planned so that hazardous cargo is separated according to the segregation rules. Where conflict with the segregation table does not occur, hazardous cargo should be stowed on deck and away from crew accommodation. The effect that loading hazardous cargo has upon TEU utilisation should be minimised. Placement of hazardous or special cargo may make some slots unacceptable stowage locations for other cargo types and care must be taken to prevent this from happening. Access to some containers (e.g. hazardous) may be required during a voyage and these should be stowed accordingly. (In most cases this means on deck.)

Intact stability [3] is constrained by guidelines set down by the Classification Society [19]. Placement of containers along the ship affects weight distribution and, as a consequence causes stress. To minimise torsion stresses, cargo must be stowed evenly across the vessel. The vessel must operate as close to zero trim as possible. If zero trim is unattainable, stern trim is preferred to bow trim so that propeller immersion is maintained and slamming force is reduced [3]. The cargo weight distribution should be within acceptable bounds set by metacentric height (GM) requirements, dead-weight limits, draft restrictions, and hull strength limitations. Ballast is used to correct stability problems, minimise torsion and shear forces and bending moment stress. However, ballast should be minimised since the vessel is in effect carrying dead weight, which directly affects its efficiency.

As a result of this diversity of factors influencing the stowage planning of containers the problem of determining a pattern of stowage that is close to optimal, whilst meeting all stowage constraints, is complex. Even over a few ports the determination of the optimum allocation of specific containers to stowage locations is computationally explosive and is not solvable in a realistic length of time. An alternative method for solving the container stowage problem, developed by the authors, is presented in this paper. In this approach, all characteristics of the problem are considered, but optimality is not necessarily sought.

3. Literature survey

Researchers, drawn from academic and commercial shipping organisations, have examined the stowage-planning problem since the 1970s. Those methods developed have been grouped into the following five main classes: simulation based upon probability, heuristic driven, mathematical modelling, rule-based expert systems, and decision support systems [4]. None of these approaches have provided a solution to the complete stowage-planning problem. A brief review of relatively recent research into automating stowage planning follows.

The first class includes the work completed by Shields [1]. Here a small number of stowage plans are created, which are then evaluated and compared by simulation of the voyage across a number of legs. The order in which loading heuristics are applied is determined using a weighted random selection procedure and this generates a limited number of different solutions. The second class of automated planning processes incorporates human planners' experience encoded in the form of heuristics. This class includes the work completed by Martin [5] automating stowage planning at container-terminals. These heuristics can produce a complete, but rarely near-optimum, solution to the container-terminal stowage problem without the interaction of a user. The third class includes work carried out by Botter [4] and Cho [6] exploring the application of mathematical models and linear programming to the problem. Those practising this method of solving the stowage problem have incorporated too many simplification hypotheses, which have made their approaches unsuitable for practical applications. The fourth class explores the potential of applying the theory of artificial intelligence to cargo stowage problems. This class includes the work of Dillingham [7], Perakis [8], Wilson [9–11] and Sato [12]. The work included within the, fifth and, last class is entirely separate to the rest. No effort is made here to automate the generation of stowage solutions. Instead, sets of tools are made available to the users that *assist* in the generation of stowage solutions. The works of Saginaw [13], Lang [14] and Sansen [15] belong to this class. The partnership of stowage planning tools and human expertise has, to date, provided the best commercial improvements.

4. Planning methodology

This section describes the stowage planning model, overviews of the model's underlying data-structures and the planning processes, along with a summary of stowage objectives and their corresponding mathematical formulation within the planning methodology. Intact stability constraints are well documented in existing literature [3] and are omitted from the following discussion of the planning methodology. Emphasis is given to the underlying heuristics used to generate stowage solutions and their subsequent evaluation. For the voyage considered:

- at each POD, unloading and loading occurred, but the latter did not begin until the former had finished;
- two cranes were available for loading and unloading at each POD.

Given the computational difficulties associated with

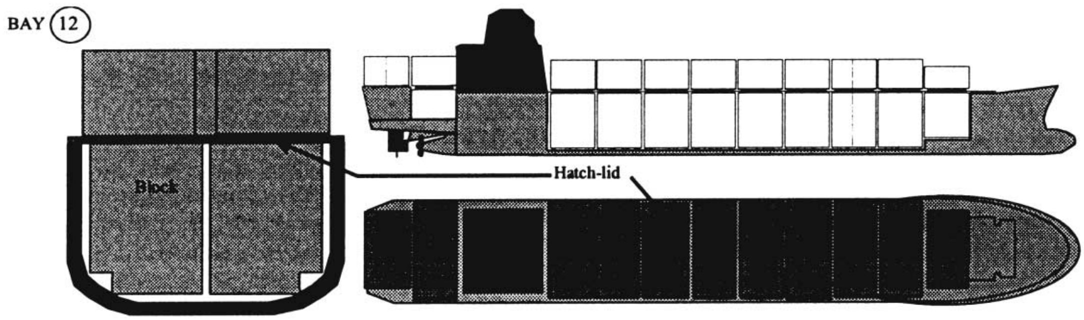

Fig. 2. Blocked container ship abstraction.

producing an exact solution for the stowage problem, it was necessary to decompose the planning process into two sub-processes [9–11] that modelled the human planner's approach, namely:

1. A *strategic planning process*, where generalised containers are assigned to a blocked cargo-space in which slots corresponding to hatch-lids are grouped together (illustrated in Figs. 2 and 3).
2. A *tactical planning process*, where specific containers are assigned to specific slots within blocks determined during the strategic planning phase (illustrated in Fig. 4).

The strategic planning process generates a generalised cargo stowage distribution. This models human planners' use of documents called the *General Arrangement* and the *Outline Plan* [16] to plan stowage, and reduces the combinatorial size of the problem whilst retaining its inherent characteristics [9–11].

Cargo can be seen as having a specific relationship to hatch-lids, which are the removable separators of above-deck and below-deck cargo, and are usually composed of a number of sections that interlock latitudinally. In particular, above deck cargo can be placed across two sections of the lid (see Fig. 4). This allows the grouping of locations into blocks of cargo locations that have both the same longitudinal position (indicated in Fig. 2) and a partnership relationship with these sections of hatch-lids. This has consequences for which lids and containers must be removed by cranes to allow access to other containers and locations.

Blocking the cargo-space of the container-ship enables the number of options for specifying container placements available at any stage of the planning process to be reduced from, perhaps, thousands of possibilities to within a hundred.

Now, the problem is reduced to allocating specific containers within a *part* of the container ship (a block) [9–11]. In a second, tactical planning, phase, the exact slot occupied by each container at the current port-of-call is determined. The combinatorial difficulties associated with attempting to make specific placements within the entire cargo space and avoided by this two-stage process.

Each block is composed of a number of locations in the same hatch (latitudinal grouping), shown in Fig. 4. This procedure models the human planner's conceptual approach and their use of documents called *Bay Plans* [16].

4.1. Strategic planning phase

Here, the underlying representation for the blocked cargo-space and the formulation of an objective function that measures how well a stowage arrangement meets these objectives, are given.

4.1.1. Strategic stowage objectives

The objectives of the strategic planning phase are to:

- minimise the number of cargo spaces occupied by each destination;

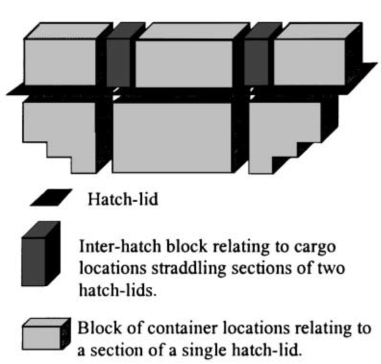

Fig. 3. Example of cargo-space blocking relating to a single hatch.

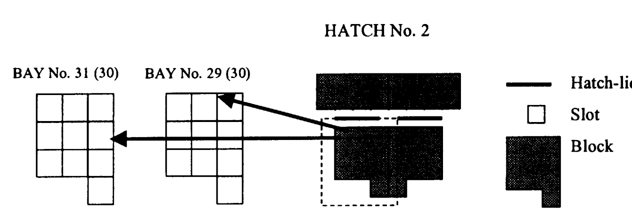

Fig. 4. Relationship between blocks and slots.

- maximise the number of cranes in operation at each POD.
- minimise the number of hatch-lids moved;
- minimise the number of re-handles;
- minimise the number of cargo blocks occupied by containers.

4.1.2. Underlying model and definitions

The objective function used to evaluate solutions to the strategic planning problem requires a number of definitions that model the underlying structure of the problem. These are shown below, employing Z notation [20], as sets and functions applied to sets. (In these, dom is *domain*, ran is *range*, # is set cardinality or size.)

- C: $\{c_1 \ldots c_{\mathrm{nc}}\}$ is the set of all containers;
- nc is the number of containers;
- P: $\{p_1 \ldots p_{\mathrm{nd}}\}$is the set of all POD;
- nd is the number of POD;
- S: $\{s_1 \ldots s_{\mathrm{ns}}\}$ is the set of all stowage locations;
- ns is the number of stowage locations;
- D: P → PC is each set of containers associated with each destination;
- $H : \mathbb{N}_1 \rightarrow$ PC is each set of containers associated with each hatch;
- nh: #{dom H} is the number of hatches;
- $B : \mathbb{N}_1 \rightarrow$ PC is the set of containers associated with each block;
- $R : \mathbb{N}_1 \rightarrow$ PC is the set of blocks associated with each corresponding upper-block;
- nr: #{dom R} is the number of upper-blocks;
- $L : \mathbb{N}_1 \rightarrow$ PS is the set of containers stowed under each hatch-lid;
- nl: #{dom L) is the number of lids;
- max: $\mathbb{N}_1 \rightarrow \mathbb{N}_1$ is a function that returns the capacity of a block;
- vol: $\mathbb{N}_1 \rightarrow \mathbb{N}$ is a function that returns the volume of used space within a block;

4.2. The strategic objective function

The objective function used to evaluate solutions to the longitudinal stowage problem examines a stowage pattern in nine ways. Its general expression is

$$f = (f_1 * 2) + (f_2 * 1) + (f_3 * 1) + (f_4 * 4) + (f_5 * 3) + (f_6 * 10) + (f_7 * 2) + (f_8 * 4) + (f_9 * 3), \tag{1}$$

where f_i and its weight represent, respectively, an abstracted measure of one factor of the attractiveness of a solution and its relative importance. A low value of f indicates a good solution.

One set of terms of the objective function concerns the production of good block stowage, which in turn brings about efficient hatch-lid movement. The first of these terms, f_1, counts the number of hatches occupied by containers of each POD

$$f_1 = \sum_{i=1}^{\mathrm{nd}} \sum_{j=1}^{\mathrm{nh}} \begin{pmatrix} 1 & \text{if } (\exists c : C \mid c \in (\mathrm{ran}(d_i) \cap \mathrm{ran}(h_j))), \\ \text{else} & 0. \end{pmatrix}. \tag{2}$$

Secondly, f_2 counts the number of POD that exist within each hatch

$$f_2 = \sum_{i=1}^{\mathrm{nh}} \sum_{j=1}^{\mathrm{nd}} \begin{pmatrix} 1 & \text{if } (\exists c : C \mid c \in (\mathrm{ran}(d_i) \cap \mathrm{ran}(h_j))), \\ \text{else} & 0. \end{pmatrix}. \tag{3}$$

Then f_3 counts the number of blocks occupied by containers of each POD. Minimising these three terms ensures good block stowage

$$f_3 = \sum_{i=1}^{\mathrm{nd}} \sum_{j=1}^{\mathrm{nb}} \begin{pmatrix} 1 & \text{if } (\exists c : C \mid c \in (\mathrm{ran}(d_i) \cap \mathrm{ran}(b_j))), \\ \text{else} & 0. \end{pmatrix}. \tag{4}$$

A second set of terms measures whether effective crane usage is possible, with low values reflecting such good stowage. The term f_4 counts how many hatches are occupied by containers of each POD and then compares this with how many cranes there are at that POD (in this case 2). Ideally, the number of cranes at a given POD should be a factor of the number of hatches occupied by that POD.

$$f_4 = \sum_{i=1}^{\mathrm{nd}} \left(\mathrm{mod} \left(\sum_{j=1}^{\mathrm{nh}} \begin{matrix} 1 & \text{if } (\exists c : C \mid \in (\mathrm{ran}(d_i) \cap \mathrm{ran}(h_j))) \\ \text{else} & 0 \end{matrix} \right) 2 \right). \tag{5}$$

A good spread of containers between hatches allows all cranes to be used simultaneously throughout the unloading process, as reflected by, f_5

$$f_5 = \sum_{i=1}^{\mathrm{nd}} \mathrm{ABS}\left(\left(\mathrm{Max}\begin{pmatrix} \forall_j : 1 \ldots \#\mathrm{nh} \mid \mathrm{count} = \Diamond\bullet \\ \mathrm{count} = \mathrm{count}\#\{c : C \mid d_i \cap h_j\} \end{pmatrix} \right) - \left(\sum_{j=1}^{\mathrm{length(count)}-1} \left(\begin{matrix} \begin{pmatrix} \forall_j : 1 \ldots \#\mathrm{nh} \mid \mathrm{count} = \Diamond\bullet \\ \mathrm{count}' = \mathrm{count} \wedge \#\{c : C \mid d_i \cap h_j\} \end{pmatrix} - \\ \left(\mathrm{Max}\begin{pmatrix} \forall_j : 1 \ldots \#\mathrm{nh} \mid \mathrm{count} = \Diamond\bullet \\ \mathrm{count} = \mathrm{count}\#\{c : C \mid d_i \cap h_j\} \end{pmatrix} \right) \end{matrix} \right) \right) \right). \tag{6}$$

Next, f_6 penalises stowage patterns in which containers of a particular destination are stowed inside two hatches and those hatches are adjacent (preventing the two cranes from working simultaneously)

$$f_6 = 1 \text{ if } \left(\begin{array}{c} \forall_i :\ 1\ldots\#\mathrm{nd} \mid \forall_j :\ 1\ldots\#\mathrm{nh} \mid \exists c :\ C \mid \left(\sum_{k=1}^{\mathrm{nh}} (1 \text{ if } \exists c :\ C \mid c \in (\mathrm{ran}(d_i) \cap \mathrm{ran}(h_k)) = 2) \right) \\ \wedge c \in ((\mathrm{ran}(d_i) \cap \mathrm{ran}(h_j)) \cap (\mathrm{ran}(d_{i+1}) \cap \mathrm{ran}(h_{j+1}))) \end{array} \right). \tag{7}$$

fewest available legal stowage locations first. Within the groups of different types of containers, those groups with the furthest POD are placed first in sequence. The fitness of the solution reflects an abstract measure of the

Lastly, the spread of containers over the removable hatch-lids also provides a measure of crane efficiency, as reflected by f_7

$$f_7 = \sum_i^{\mathrm{nc}} \sum_j^{\mathrm{nd}} \sum_k^{\mathrm{nl}} \begin{pmatrix} 1 & \text{if } c_i \in (\mathrm{ran}(d_j) \cap (\mathrm{ran}(l_k))) \\ \text{else} & 0 \end{pmatrix}. \tag{8}$$

The third, and final, set of terms measures container rehandles and overstows, with low values of the terms indicating low numbers of these undesirable movements. The term f_8 counts how many containers are stowed on hatch-lids, beneath which are containers destined for an earlier POD

$$f_8 = \sum_{i=1}^{\mathrm{nb}} \sum_{j=1}^{\mathrm{nb}} \sum_{k=1}^{\mathrm{nc}} \sum_{l=1}^{\mathrm{nc}} \times \begin{pmatrix} 1 & \text{if } (c_k \in (\mathrm{ran}\ b_i)) \wedge (c_l \in (\mathrm{ran}\ b_j)) \wedge \\ & ((\mathrm{dom}\ r_j) \in (\mathrm{ran}\ r_i)) \bullet D^{\sim}[c_l] > D^{\sim}[c_k] \\ \text{else} & 0 \end{pmatrix}. \tag{9}$$

Also, f_9, counts how many empty spaces exist below a hatch-lid that supports containers. These spaces are unavailable without first removing the hatch-lid and any containers stowed on it, and therefore indicate of poor stowage

$$f_9 = \sum_{i=1}^{\mathrm{nr}} \sum_{j=1}^{\mathrm{nr}} \begin{pmatrix} (\max(r_i) - \mathrm{vol}(r_i)) \\ \text{if} \begin{pmatrix} (r_j \in \mathrm{ran}(r_i)) \wedge (\mathrm{vol}(r_j) > 0) \\ \wedge (\mathrm{vol}(r_i) < \max(r_i)) \end{pmatrix} \end{pmatrix}. \tag{10}$$

4.3. Implementation using branch and bound search

The branch and bound approach to search is a very useful method for solving discrete optimisation, combinatorial optimisation and integer problems in general [17]. For the blocked stowage problem, the Branch and Bound algorithm and related sub-procedures are specialised as follows.

1. *Initialisation*. The initial state is made up of the cargo-space, an ordered list of all containers to be loaded at the current port of call and a fitness value of the stowage arrangement. The cargo-space is composed of a list of areas to fill that correspond to blocks within the ship. The list of containers to be loaded has containers with the fewest available legal stowage locations first. Within the groups of different types of containers, those groups with the furthest POD are placed first in sequence. The fitness of the solution reflects an abstract measure of the cost, based upon simulation of the unloading process at PODs.
2. *Branching*. New solutions are generated that reflect placements of the first container in the load-list within the cargo-space of a partial-solution. All invalid solutions are then removed from the list of new states. If after expanding a partial solution a feasible solution for the longitudinal stowage problem is found, then it is set aside.
3. *The search strategy*. The candidates produced during the branching process are ordered according to the least fitness value determined by the objective function and the least number of containers remaining within its associated load-list. This new list is placed at the front of the existing list of partial-solutions. This strategy reflects a depth first approach to the search process.
4. *Pruning*. When one candidate has the same, or worse, fitness value as another but has more containers to load then it can be deleted from the pool of partial solutions.
5. *Choice of new sub-problem*. The partial-solution with the best fitness value is selected as the new current candidate problem and the algorithm continues in a similar manner until n solutions are found and d destinations are processed. Upon delivery of n candidates the search process for the current port of call is terminated, the problem is reinitialised, and the process repeated again for each of the n solutions at the next POD.

This process simulates a planning procedure at a given number of destination ports. Once this process has been repeated for each destination, the best solution is the one with the least summation of the fitness values accumulated at each port.

5. Tactical planning phase

In this phase, the best, generalised, long-term solution determined during the strategic planning phase is refined. Here, the stowage objectives followed when making short-term stowage decisions are presented, along with the underlying representation for the cellular cargo-space and the formulation of an objective function that measures how well a stowage arrangement meets these objectives.

5.1. Tactical stowage objectives

Planners employ a variety of generalised and specialised stowage to direct the placement of containers [9]. For the model under consideration, the following are considered salient:

- re-handles are to be minimised;
- container weight is to be graded upwards, heaviest to lightest;
- stacks of containers) with mixed POD are to be minimised.

5.2. Underlying model and definitions

The objective function used to evaluate solutions to the tactical planning problem requires a number of definitions that model the problem's underlying structure, specifically:

- I:$\{c_1 \dots c_{nc}\}$ is the set of all containers;
- D_i is the destination port of container i;
- DR_i is the set of restows related to container i;
- DW_i is the set of containers in the same stack stowed above container $i$with a greater weight;
- DS_i is the set of containers stacked with container i with a different POD.

5.3. The tactical objective function

The general expression for the objective function for the problem of container assignment within a block is

$$f = (f_{10} * 3) + (f_{11} * 1) + (f_{12} * 2), \tag{11}$$

where f_i and its weight represent, respectively, an abstracted measure of one factor of the attractiveness of a solution and its relative importance. Better solutions will return lower objective function values.

The first term of the objective function, f_{10}, counts the number of restows

$$f_{10} = \sum_{i=1}^{nc} \sum_{j=1}^{nc} \begin{pmatrix} 1 & \text{if } j \in DR_i \\ \text{else} & 0 \end{pmatrix}. \tag{12}$$

The second term of the objective function, f_{11}, counts the number of containers with a different POD stowed in the same stack

$$f_{11} = \sum_{i=1}^{nc} \sum_{j=1}^{nc} \begin{pmatrix} 1 & \text{if } j \in DS_i \\ \text{else} & 0 \end{pmatrix}. \tag{13}$$

The third term of the objective function, f_{12}, examines weight gradation by counting the number of containers with a greater weight stowed above each other in the same stack

$$f_{12} = \sum_{i=1}^{nc} \sum_{j=1}^{nc} \begin{pmatrix} 1 & \text{if } j \in DW_i \\ \text{else} & 0 \end{pmatrix}. \tag{14}$$

5.4. Optimisation using Tabu search

Tabu search can be viewed as an iterative technique that explores a set of problem solutions by repeatedly making moves from one solution s to another solution s' located in the neighbourhood $N(s)$ [18]. For the container stowage problem, s is the stowage configuration for the entire container ship and $N(s)$ is the set of all configurations obtained by making moves within a single hatch, with each hatch being optimised separately. These moves are performed with the aim of reaching a near optimal solution by the evaluation of some objective function $f(s)$ to be minimised. To prevent the search process from returning a local optimum f, a guidance procedure is incorporated that accepts a move from s to s' even when $f(s') > f(s)$. Should no improving move be found in a given number of iterations then the original, best, local solution is returned as the global solution. This in itself could lead to cycling causing the process to return repeatedly to the same local solution without moving towards a global solution. Tabu search circumvents the problem of cycling by preventing recent moves from reoccurring for a given number of iterations. For each solution s, m is a set of, legal, non-tabu moves which can be made to obtain a new solution s'. ($s' = S \oplus m | N(s) = s' | \exists m \in M(s)$). For the assignment problem of container within a block, the neighbourhood $N(s)$ was determined by the blocked stowage procedure. An initial random value for s would suffice, but the application of a packing heuristic that generates a sensible value for s ultimately improves the efficiency of the search algorithm [9]. The form of the procedure used to optimise the arrangement of containers is as follows:

$$s* := f(s), \qquad k := 1, \qquad j := 1$$

While ($j <$ Max (j)) and ($k <$ Max (k)) and ($f(s*) \neq 0$)
  $j := j + 1$;
  $M* \subseteq N(s,k)$ (all legal, non-tabu, states);
  Choose the best s' in $M*$ and set $s := s'$;
  If $f(s') < f(s*)$ then $s* := s'$, $k := 1$ else $k := k + 1$.
End of While

The form of Tabu search described is a very simple version. The pre-planning that occurs during the strategic phase prunes the search space significantly, resulting in subproblems where optimal solutions can be found easily. Consideration given to aspiration levels, intermediate or long term memory, and other features described in the literature [18] proved unnecessary.

6. Computational experiments

Results were obtained on a 166 MHz Pentium with 40 Mb of memory using Allegro Lisp to encode the blocking and GFA (a PC-based 3GL with a high degree of functionality and graphic display features) to encode the specific placement

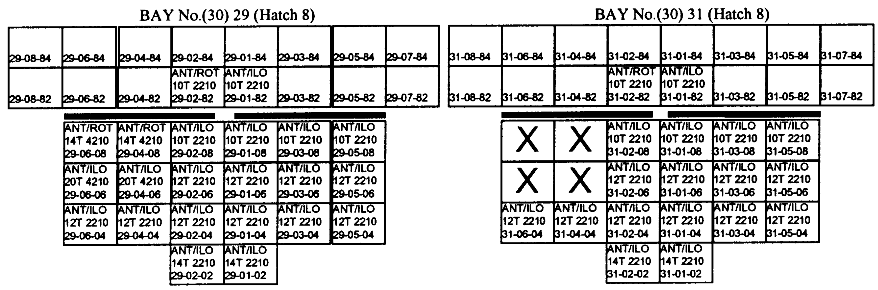

Fig. 5. Bay plan giving container POD, origins, types and weights.

algorithm. A generalised solution to the described problem was obtained in approximately 90 min whereas specialised solutions for all blocks were produced in under an hour. The state space size for the strategic planning phase will vary with vessel capacity and number of ports considered, but the blocking of cargo-space is believed to ensure that solutions of acceptable quality can always be generated in a viable time.

The application of Tabu Search during the tactical phase requires little processing time, as it involves the placement of fewer than 100 TEUs (and typically cargo blocks will hold approximately 12–60 TEUs). Below-deck blocks have restrictions on container lengths, and experimentation on typical loads generated optimum solutions in as few as 15 iterations, and a recency list of just two moves. For above-deck blocks, this number increases due partly to variations in container lengths, but mostly due to the increased likelihood of hazardous cargo segregation requirements. However, in the worst cases, no more than 200 iterations, and recency lists of up to seven moves are required.

The commercially sensitive nature of real data used, preclude further meaningful quantitative analysis of the results of this approach here. However, qualitatively, and from a knowledge-engineering perspective, the authors observe that the solutions obtained in experiments meet all constraints and reflect the stowage objectives described. The plans generated being reported by industry experts consulted as comparable with those of human planners. Further, the automated approach outlined in this paper allows consideration of more stowage plans in the time available for planning than human planners can manage.

This procedure was used to generate a complete set of Bay Plans, optimised with respect to cargo allocated in the strategic phase (an example of which is shown in Fig. 5). Note that in Fig. 5: X marks the tail end of a 40′ container; ROT is an earlier POD than ILO; containers numbered 4210 are 40′ in length and ones labelled 2210 are 20′ in length — hence the four 40′ containers are lighter 'per foot' than the 20′ containers below them.

7. Conclusions

Through a comprehensive knowledge engineering exercise, a model has been constructed for solving the container-ship loading problem that takes advantage of how human planners solve this problem. The implementation of the model described in this paper allows sub-optimal solutions to the problem to be determined without requiring the intervention of a human planner. Moreover, the solutions are obtained in a reasonable amount of processing time using available computer software and hardware.

References

[1] J.J. Shields, Container-ship stowage: a computer-aided preplanning system, Mar. Technol. 21 (4) (1984) 370–383.

[2] Private communication with Maritime Computer and Technical Services.

[3] L.L. Goldberg, Intact stability, in: J.P. Comstock (Ed.), Principles of Naval Architecture, Architectural Marine Engineer, New York, 1980 (chap. II).

[4] R.C. Botter, M.A. Brinati, Stowage container planning: a model for getting an optimal solution, IFIP Trans. B (App. in Tech.) B-5 (1992) 217–229.

[5] G.L. Martin, S.U. Randhawa, E.D. McDowell, Computerised container-ship load planning: a methodology and evaluation, Comput. Ind. Engng. 14 (4) (1988) 429–440.

[6] D.W. Cho, Development of a methodology for containership load planning, PhD thesis, Oregon State University, 1984.

[7] N. Perakis, J.T. Dillingham, The Application of Artificial Intelligence Techniques in Marine Operations, Ship Operations, Management and Economics International Symposium U.S.M.M.A., Kings Point, New York, 1987.

[8] J.T. Dillingham, A.N. Perakis, Application of Artificial Intelligence in the Marine Industry, presented at Fleet Management Technology Conference, Boston, USA, 1986.

[9] I.D. Wilson, The application of artificial intelligence techniques to the deep-sea container-ship cargo stowage problem, PhD thesis, University of Glamorgan, 1997.

[10] I.D. Wilson, P.A. Roach, The deep-sea container-ship stowage problem: modelling and automating the human planning process, in: Proceedings of the Second International ICSC Symposium on Intelligent Industrial Automation, ICSC, Canada/Switzerland, 1997, pp. 129–135.

[11] I.D. Wilson, P.A. Roach, Principles of combinatorial optimisation applied to containership stowage planning, J. Heuristics 5 (4) (1999) 403–418.

[12] K. Sato, H. Itoh, Y. Awashima, Expert system for oil tanker loading/unloading operation planning, in: C. Barauna Vieira (Ed.), Computer Applications in the Automation of Shipyard Operation and Ship Design, VII, Elsevier, North-Holland, Amsterdam, 1992.

[13] D.J. Saginaw, A.N. Perakis, A decision support system for container-ship stowage planning, Mar. Technol. 26 (1) (1989) 47–61.
[14] G.J.P. Lang, Some computer aids in the loading of deep sea container vessels — a personal experience, in: Computer Applications in Operation and Management of Ships and Cargoes, London, 1985.
[15] H. Sansen, Ship-planner, a conception support tool for the bay plans of container ships, Systemia, Domaine de ST Hilaire, Pichaury 13290 AIX LES MILLES, France, 1989.
[16] D.K. Roach, B.J. Thomas, Portworker development programme Unit C.2.2, Container-ship Stowage Plans, International Labour Organisation, Geneva, ISBN 92-2-109271-2, 1994.
[17] B. Cao, Transportation problem with nonlinear side constraints a branch and bound approach, ZOR — Meth. Models Oper. Res. 36 (1992) 185–197.
[18] F. Glover, A user's guide to tabu search, Ann. Oper. Res. 41 (1993) 3–28.
[19] Stowage and Segregation Guide to IMDG-Code, K.O. Storck Verlag, Hamburg, 1998.
[20] J.M. Spivey, The *Z* Notation: A Reference Manual, Prentice-Hall, UK, 1989.