Proceedings of the 2010 Winter Simulation Conference
B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, eds.

AUTOMATED STOWAGE PLANNING FOR LARGE CONTAINERSHIPS WITH IMPROVED SAFETY AND STABILITY

Min Zeng
Malcolm Yoke Hean Low
Wen Jing Hsu
Shell Ying Huang
Fan Liu
Cho Aye Win

Nanyang Technological University
Parallel and Distributed Computing Centre, School of Computer Engineering,
Nanyang Technological University, Nanyang Avenue,
Singapore, 639798, SINGAPORE

ABSTRACT

Stowage planning for container ships is a core activity of shipping lines. As the size of containership increases, generating a stowage plan with good safety and stability for a large containership becomes increasingly difficult. In this paper, we present an automated stowage planning system for large containerships which consists of three modules: the stowage plan generator, the safety and stability adjustment module, and the optimization engine. This paper focuses on the safety and stability adjustment module which resolves the stability issues of a stowage plan by adjusting the distribution of container weights by stowing containers in alternative feasible locations and fine-tuning stability parameters through adjusting the ballast in tanks onboard. Using shipping data for a large 7000 TEUs containership on a multi-port voyage, we demonstrate that our system can generate stowage plans with improved safety and stability compared to those generated by experienced planners.

1 INTRODUCTION

Stowage planning for container ships is a core activity of shipping lines. It is a difficult problem to solve because of the combinatorial nature of alternative mappings from the containers to the stowage locations on a ship and the numerous constraints associated with the ships and the types of containers. Although much research work has been devoted to this problem, most existing approaches target to minimize the loading time of all containers (Ambrosino 2004) or the number of shifts (Avriel 1998) rather than weight distribution.

Imbalance in weight distribution of containers onboard a ship can cause ship stability problems and lead to disasters. Currently, the allocation of containers and hence their weight distribution to slots in a stowage plan is still carried out manually by human planners based on their experience. With the capacity of containerships reaching ten thousand TEUs (Twenty Foot Equivalent Unit) and more, it is increasingly difficult for a planner to manually generate a feasible stowage plan that conforms to the various physical constraints of the containers to be loaded, and at the same time achieving a good weight distribution of containers across the ship.

978-1-4244-9864-2/10/$26.00 ©2010 IEEE

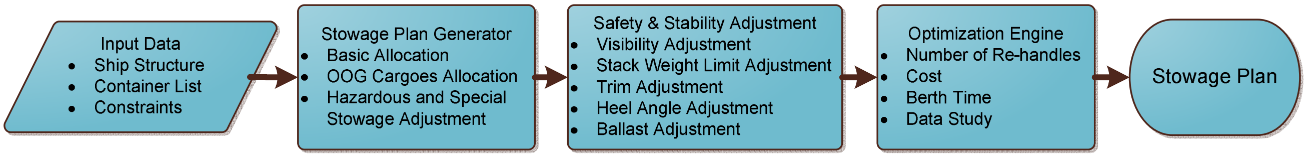

Figure 1: Systems of Automated Stowage Planning

Although the stability issues of containership stowage plan can be resolved to a certain extent through rearranging containers in the ship, there are inherent constraints to the number of possible stowage locations for each container due to factors such as size, type, weight and the destination of the containers. To achieve the desired stability needed for safety during sailing, any remaining stability issues after the container adjustment stage need to be resolved by adjusting the amount of ballast in tanks onboard the ship. Thus, ballast adjustment also plays a vital role in generating stowage plan with good ship stability. Moreover, the amount of ballast directly affects the cost of shipping operations. Therefore, it is also necessary to minimize the ballast. However, currently, ballast adjustment is also carried out manually during stowage planning by human planner depending on their experience.

The objective of our study is to develop a fully automated system for stowage planning of large containerships. Figure 1 shows the framework of our design for an automated stowage planning system. The input to the system consists of a list of containers for loading and unloading at each port on a multi-port voyage. The stowage planning process has 3 stages: (1) the *stowage plan generator* produces an initial stowage plan which fulfils a set of constraints without the consideration of ship stability; (2) the *stability module* checks the stability of the initial stowage plan and adjusts it to satisfy the stability requirements of the ship; (3) the *optimization engine* takes this feasible stowage plan and optimizes it based on specific objectives (such as minimizing the number of re-handles). As the work related to the stowage plan generator module has been described in (Xiao 2009), in this paper, we only present our work on stability adjustment using container weights and ballast. The work with reference to the optimization engine is still in progress and will be described in our future publication.

The paper consists of six sections. The next section is concerned with the review of related literatures. Section 3 describes the basic structure of the ship in detail. Section 4 presents the main constraints of the stability conditions and our proposed algorithmic approach. In Section 5, we give a case study on a large containership and present some experimental results aimed at validating the proposed approach. Section 6 concludes the paper and outlines some future work.

2 LITERATURE RRVIEW

Since the 1970s, the container stowage planning problem has been studied by shipping lines and researchers. The existing research is mostly focused on the container loading problem, which can be formulated as a combinatorial optimization problem (Aslidis 1989 and Wilson 2001). The size of the solution space for the container stowage planning problem depends on the ship capacity and the shipping demand at each port. Even for a medium size containership, the problem is nontrivial due to the large number of variables. Moreover, the problem has been proved to be NP-hard, which implies that it is very unlikely to guarantee finding an optimal solution in a reasonable processing time (Ambrosino 2004). Meanwhile, several researchers developed heuristic-based computerized methodologies to provide workable solutions to stowage planning. A brief review of some recent research follows.

The early study about the container stowage problem can be traced back to the work by Aslidis in 1989 and 1990, who examined the stack overstowage problem of small size problem under certain assumptions (containers have same type, same weight etc.). Aslidis's work led to a set of heuristic algorithms which were used to solve the container loading problem without stability consideration. Another early work was carried out by Imai and Miki (1989) who considered the minimization of the loading-related re-handles. They formulated the problem as an integer programming problem with one objective function that minimizes the expected number of containers to shift. The contribution rate for Gravity Metacentric (GM) is solved by the algorithm which consists of two solution methods, with the classical assignment problem solved by the Hungarian method and the integer programming by branch-and-bound. However, in their approach, they only considered one metric, *GM*, in the ship stability issue. Other factors such as *heel angle* and *trim* were not considered. This simplified assumption makes their approach not applicable to solving real world stowage problems.

Avriel and Penn (1993) formulated the stowage planning problem into a 0-1 binary linear programming. They found that the general algorithm is too slow even with some pre-processing of the data. Avriel et al. (1998) developed a heuristic procedure called the *suspensory heuristic* procedure with the objective of minimizing the number of container re-handles. However, they assumed that the ship has only a large cargo bay, and did not consider the issues of hatch covers and stability. Also, Avriel et al. (2000) showed that the stowage planning problem is NP-complete by showing that the stowage problem is related to the circle graphs coloring problem which is known to be NP-hard.

Wilson and Roach (1999, 2000) developed a methodology for computerising stowage planning. Their methodology embodies a two-stage process. Firstly they used branch-and-bound algorithms to assign general containers to blocks in a bay in a vessel. In the second step they used a tabu search algorithm to assign locations for specific containers. Wilson et al. (2001) presented a computer system for generating solutions to the stowage pre-planning problem based on the ship with the capacities of more than 4500TEUs using a genetic algorithm approach. However, their approach to generate a stowage plan still requires nearly 90 minutes even without the optimality guarantee.

Dubrovsky et al. (2002) used a genetic algorithm technique for minimizing the number of container movements of the stowage planning process. The authors developed a compact and efficient encoding of solutions to reduce the search space significantly. However, the authors only considered the ship to have a small, single bay, and they also ignored the stability issue which is very critical in stowage planning.

In the papers by Ambrosino et al. (1998, 2004 and 2006), the stowage planning problem is referred to as the Master Bay Plan Problem (MBPP). Ambrosino and Sciomachen (1998) reported the first attempt to derive the constraints (Ambrosino 1989) related to the nature of containers and ship locations for determining good container stowage plans, where a constraint satisfaction approach is used to define the space of feasible solutions. Ambrosino et al. (2004) described a 0-1 linear programming model for MBPP. They presented an approach consisting of heuristic pre-processing and pre-stowing procedures that allow the relaxation of some constraints of the exact model. Ambrosino et al. (2006) presented a three-phase algorithm for MBPP, which is based on a partitioning procedure that splits the ship into different portions and assigns them to containers on the basis of their destinations. However they assumed that the ship starts its journey at a port and visits a given number of other ports where only unloading operations are allowed, which implies the loading problem can only be considered at the first port.

Xiao et al. (2009) proposed a heuristic algorithm to solve a real world containership stowage problem for a 5000 TEU ship with containers loading and unloading over a voyage that consists of 8 ports. In their algorithm, the authors introduced a workload imbalance tolerance factor to quantify the distribution of container loading/unloading workload of a stowage plan from the perspective of quay cranes at the respective ports. By setting the workload imbalance tolerance factor to a suitable value, the algorithm can generate a stowage plan that has a good tradeoff between the number of containers re-handles and the utilization rate of cranes. Both of which are important objectives of MBPP. However in this work, the authors also did not address the issue of stability in the stowage plan.

Since all the research mentioned above made simplistic assumptions and few considered the stability problem, these solutions cannot be applied by the shipping lines in real life especially for large containerships. In this paper, we describe an algorithm that improves the ship stability of stowage plans generated by our stowage plan generator. The algorithm is able to consider existing containership features and constraints to rapidly generate a set of feasible plans which meet the safely sailing requirement for a containership on a multi-port voyage.

3 THE STRUCTURE OF A CONTAINERSHIP

A stowage plan consists of two main parts. The first part is the stowage details of containers onboard the ship. The second part is the ballast configuration in tanks. Before we explain our proposed safety and stability adjustment algorithm, the structures of both the container slot location and ballast tank are listed as follows.

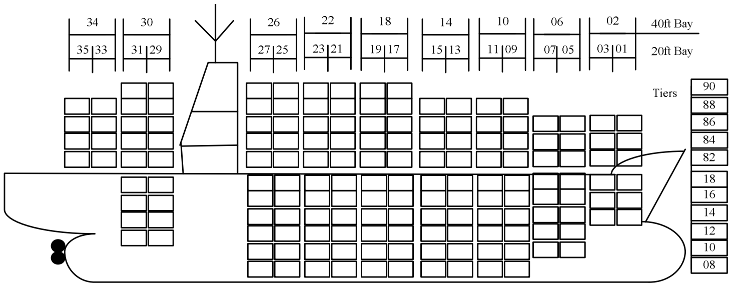

Figure 2: Cross-sectional View of a Containership

3.1 The Structure of Slot Location

The stowage planning problem is to assign a given set *C* of *n* containers with different properties to a set *L* of *m* available slot locations of a containership. The cross-sectional view of a typical containership is shown in Figure 2. A containership contains a number of bays with their IDs increased from bow to stern. There are two types of bays. A 40 foot (40') bay has an even bay ID and can be used for stowing 40' containers. A 20 foot (20') bay has an odd bay ID and can be used for stowing 20' containers. Furthermore, two adjacent 20' bays can be used as one 40' bay, such as bay14 = bay15+bay13. Each bay includes rows with row IDs numbered from centre to outside, and tiers with tier IDs numbered from bottom to top. Tiers are further divided into two sets. Tiers above deck are numbered with IDs starting from 80, and tiers below deck are numbered with IDs starting from 0. We present some definitions of ship parameters as follows:

- l_{ijk} is defined as a location in a bay with two-digit numbers i, j, k, representing the bay, row and tier of the location respectively. For instance, l_{140282} refers to the location in bay 14, row 02 and tier 82.
- t_{maxij} is defined as the maximum number of tiers in bay i, row j below deck.
- t_{minij} is defined as the minimum number tiers in bay i, row j above deck.
- x_{ijk} is defined as the distance between the centre of gravity (CG) of the ship and the location l_{ijk}.
- y_{ijk} is defined as the distance between the CG of bay i and the location l_{ijk} in the same bay.
- c_{ijk} is defined as a container which is stowed in bay i, row j and tier k.

In addition, certain properties of containers also affect the stowage planning process. In this study, we focus on the size, type, port of destination and weight of the containers:

- Size: In set *C*, there are two groups of containers, 20' containers and 40' containers respectively. For safety reason, 20' containers cannot be stowed above 40' containers.
- Type: Different types of containers can usually be stowed in a containership, such as normal containers, reefer containers, out-of-gauge containers and hazardous containers. The constraints for different container types have been considered in the stowage plan generator module. Furthermore, each type of container has its own height. The visibility effect of container height will be illustrated in Section 4.
- Port of destination: Each container has its own destination port. When the containers destined for a further port are stowed above the containers that will be unloaded at an earlier port, this scenario is

referred to as over-stow. Over-stow will cause container re-handling operations at downstream ports. In order to minimize unnecessary re-handling and re-shuffling of containers during unloading at a port, during the stowage planning process, containers going to further ports are allocated first and the containers destined for nearer ports are allocated later.

- Weight: Five categories of weight classes are defined for containers: empty, light, medium, heavy and extra heavy. The ranges of the weights are [2.5, 4), [4, 10), [10, 14), [14, 20) and [20, 30] tons, respectively. We define w_c as the weight of container *c* and w_{maxijd} as the maximum weight of the stack in bay i, row j, where the subscript *d* is 0 if the container is below deck or 1 if it is above deck. The issue of weight distribution will be further discussed in Section 4.

3.2 The Structure of Ballast Tank

Figure 3 shows the positions of the ballast tanks in a large containership. As can be seen, most of the tanks are distributed symmetrically in the horizontal direction. We present some definitions of tank parameters as follows:

- $\mathrm{Tank_i}$ is defined as the tank with ID *i*.
- w_{Tank_i} is defined as the current weight of $\mathrm{Tank_i}$.
- $w_{Tank_{maxi}}$ is defined as the maximum weight of $\mathrm{Tank_i}$.
- Lcg_i is defined as the distance between the longitudinal centre of the ship and the CG of $Tank_i$.
- Tcg_i is defined as the distance between the horizontal centre of the ship and the CG of $Tank_i$.
  - ○ $Twin_i$ is defined as a twin tank for $\mathrm{Tank_i}$. Here, two tanks with the same Lcg and opposite Tcg are defined as twin tanks.
- T is defined as a set of Tanks.

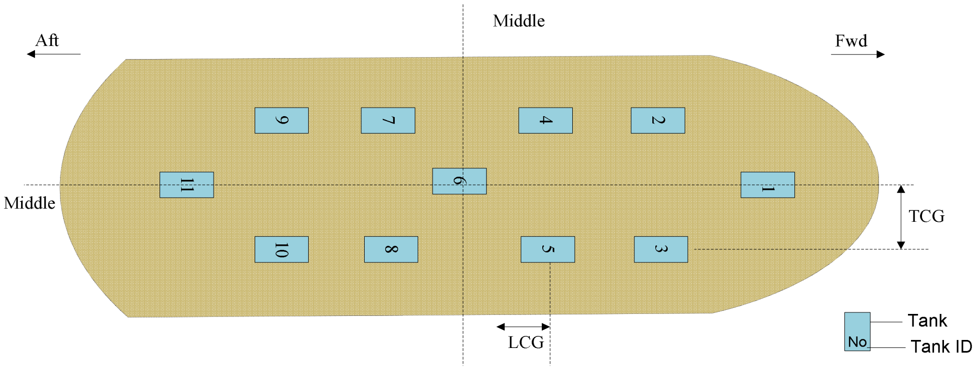

Figure 3: Ballast Tank Position Distribution

The symmetrical structure of tanks is important for ballast adjustment. With ballast tank adjustment, it is possible to achieve a better ship stability without affecting the stability result obtained in the container adjustment stage. The details for stability adjustment using ballast will be described in Section 4.

4 SAFETY AND STABILITY IMPROVEMENT ALGORITHM

As shown in Figure 1, the safety and stability module includes five procedures aimed at satisfying visibility, stack weight, trim and heel-angle requirements by adjusting stowage plan and ballast. The purpose of visibility adjustment is to make sure that the view of the sea surface from the navigation bridge is not blocked. Stack weight, trim, heel-angle and ballast are important ship stability parameters that ensure safe sailing conditions are met. The safety and stability conditions and algorithms about the visibility line limit (i.e., the maximum height limit for a special bay), stack weight limit, *trim* (i.e., the moment balance between bow and stern), *heel angle* (i.e., the moment balance between the left and right side of the ship) and ballast (i.e., the stability effect caused by the moment of ballast in tanks) are presented as follows.

4.1 The Safety and Stability Condition

The base stowage plan generated by our Stowage Plan Generator module takes into consideration the different sizes, types and destinations of containers. However, sailing safety and the weights of containers are not considered in the stowage plan generation. The safety and stability condition will be illustrated as follows.

4.1.1 The stability constraints

We develop a stability module to check the weight distribution of containers in the stowage plan and produce a feasible stowage plan that satisfies the stability conditions specified below.

$$\sum w_{c_{ijk}} \leq w_{maxijd} \text{ (If d=0, } k \leq t_{maxij}, k \geq t_{minij}) \tag{1}$$

$$-M_1 \leq \sum_{i \in A, i, k} w_{c_{ijk}} x_{ijk} - \sum_{i \in F, i, k} w_{c_{ijk}} x_{ijk} \leq M_1 \tag{2}$$

$$-M_2 \leq \sum_{j \in L, k} w_{c_{ijk}} y_{ijk} - \sum_{j \in R, k} w_{c_{i'jk}} y_{i'jk} \leq M_2, i = i' \tag{3}$$

$$-\sum_{i \in T} w_{Tank_i} Lcg_i = \sum_{i \in A, i, k} w_{c_{ijk}} x_{ijk} - \sum_{i \in F, i, k} w_{c_{ijk}} x_{ijk} \tag{4}$$

$$w_{Tank_i} Tcg_i = w_{Twin_i} Tcg_{Twin_i}, i \in T \tag{5}$$

In particular, as expressed by constraints (1), the stack weight limit safety condition requires that the total weight of containers stowed in the same stack must be smaller than the stack weight limit.

In addition, the expected value of the trim of a ship is (typically) between 0-0.5 meter, where the *Trim* is defined as the difference of the draft between the stern and bow of a ship resulting from the difference (Mom_{diff}) in between the stern moment and the bow moment. We have $trim = Mom_{diff}/(MTC * 100)$, where MTC is the moment required to produce a one centimeter change in the trim.

Since MTC is a constant, the difference of longitudinal moment between the stern and bow side must be less than $M_1 = 0.5(MTC * 100)$, as expressed by constraint (2), where A is a set of bays on the stern and F is a set of bays on the bow.

Moreover, the horizontal stability condition requires that the moment on the left and right side of a ship must not differ by more than a given tolerance M_2. This condition is expressed by constraint (3), where L is a set of rows on the left side of a ship and R is a set of rows on the right side of a ship.

Finally, as expressed by constraint (4), the ballast adjustment requires that a change in tank longitudinal moment should be equal to the difference of the longitudinal moment of the ship with all the ballast tanks being empty. Meanwhile, in order to maintain the heel angle achieved from the previous stability adjustment steps, the change in the transverse moment caused by the ballast adjustment should be zero. Therefore, we try to link the twin tanks together. In other words, the ballast level of twin tanks will be kept equal as expressed in constraint (5).

4.1.2 Line of Visibility Constraint

A clear view of the sea surface from the navigation bridge is vital for safe sailing of a containership. However, this view may be blocked by cargo or other obstructions. As shown in Figure 4, the blocked area is called the blind sector. The blind sector is limited by the range of the visibility limit sector which is the maximum area that can be blocked. In other words, containers located in front of the navigation bridge should not be stowed such that they exceed the line of visibility which links the navigation bridge with the edge of the visibility limit sector. Some definitions for the consideration of the line of visibility condition are listed as follows:

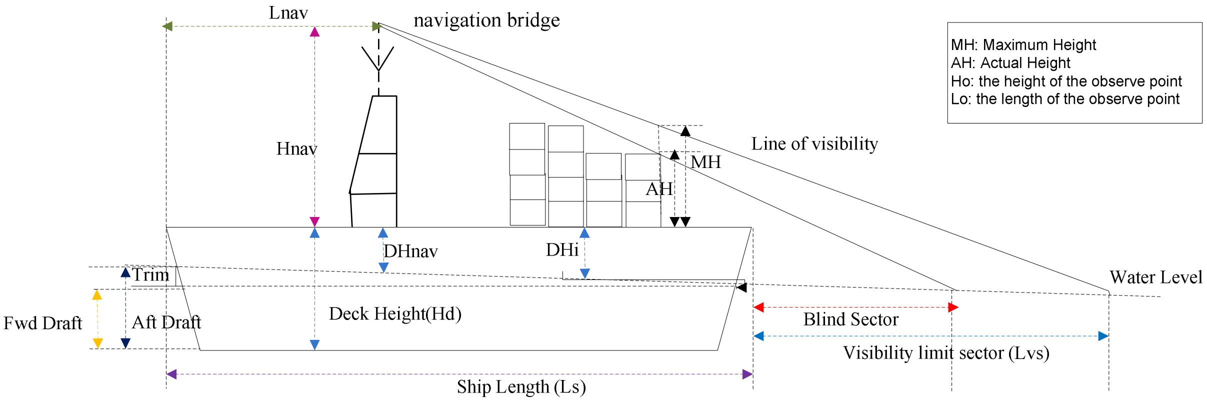

Figure 4: Line of Visibility Constraint

- Ls is defined as the longitudinal length of the ship.
- Hd is defined as the distance between the deck and the bottom of the ship.
- Hnav is defined as the distance between the deck and navigation bridge.
- Lnav is defined as the distance between the navigation bridge point and the stern side of the ship.
- Lvs is defined as the length of the visibility limit sector in front of the bow side.
- Trim is defined as the difference between the Aft draft and the Fwd draft.
- h_{ijk} is defined as the height of the container located at the location l_{ijk}.
- L_i is defined as the distance between the forward side of bay *i* and the stern side of the ship.
- $DHnav$ is defined as the distance between the deck and the water level at the navigation bridge point.
- DHi is defined as the distance between the deck and the water level at the forward of bay *i*.
- Max_i is defined as the maximum height (MH) of bay *i* above deck.

The safety module is developed to check the visibility constraint, with the safety condition illustrated as follows:

$$\max_j \left(\sum_{i>i_{nav}, k\geq t_{minij}} h_{ijk} \right) \leq Max_i \tag{6}$$

where $Max_i = (Hnav + DHnav) * \frac{Ls+Lvs-L_i}{Ls+Lvs-Lnav} - DHi$ and i_{nav} is the bay ID of the navigation bridge.

The safety condition requires that the top container in a stack must observe the visibility limit. In other words, the height of the position which the top container is located in must not be greater than the maximum height limit of the bay which the stack belongs to. As expressed in condition (6), the maximum height of any stack in a bay located in front of the navigation bridge should not exceed the maximum height limit of the bay.

4.2 Adjustment Algorithm

For safety and stability adjustment, we develop a simple yet effective heuristic algorithm. We first adjust the stack weight of containers in all the rows across the ship by moving or exchanging containers. Secondly, the visibility limit will be checked. If the view of sea surface is blocked by cargo, the illegal containers will be shifted to other empty space. Then, if required, adjustments to the stowage location of containers are carried out to achieve the desired trim by balancing the moments between stern and bow. Finally, we adjust the weight of containers in the same bay to ensure a good heel angle by balancing the moments between the left and right sides of the ship. The adjustments in each step are carried out such that the safety and stability conditions obtained by the previous steps are not affected.

4.2.1 Stack weight adjustment

The stack weight limit condition is an important safety factor for stowage planning. If the weight of the containers in one stack exceeds the stack weight limit, the stack may collapse during voyage. We define Pod_c as the destination of container c, C_{Pod_t} as a set of containers which are loaded at the current port and will be unloaded at port *t* and L_{Pod_t} as a set of locations which are occupied by containers belonging to C_{Pod_t}. We carry out the following three steps to deal with this problem.

- Exchange stage: Firstly, we select the set of stacks S_{exceed} whose stack weight limits are exceeded and the set of stacks S_{free} whose weights are less than the stack weight limit. Next we choose the heaviest container $c_{ijk} \in C_{\mathrm{Pod}_t}$ in one of the stack in S_{exceed} to swap with a lighter container $c_{i'j'k'} \in C_{\mathrm{Pod}_t}$ from one of the stack in S_{free}. A stack is moved from S_{exceed} to S_{free} if its stack weight drops below the corresponding stack limit.
- Moving stage: If there are still stacks remaining in S_{exceed} after the Exchange stage, the container $c_{ijk} \in C_{Pod_t}$at the top of one of the stack in S_{exceed} is moved to another slot location subject to certain constraints, as detailed below:
- Firstly, choose an empty slot location which satisfies the following three conditions:
    a) The empty location $l_{i'j'k'}$ should be located at the stack that does not exceed the stack weight limit, which is denoted by $l_{i'j'k'} \in S_{free}$.
    b) After loading the container c_{ijk} to the location, the total weight of the stack should be less than the stack weight limit, i.e., $w_{c_{ijk}} + \sum_k w_{c_{i'j'k'}} \leq w_{maxi'j'd}$ (if d=0, $k \leq t_{maxij}$, $k \geq t_{minij}$).
    c) The locations below the chosen location should have been filled up with containers of the same port of destination with the container c_{ijk}.
- Secondly, if no slot location satisfies all three conditions mentioned above, choose the location $l_{i'j'k'}$ which satisfies only conditions (a) and (b). The containers below the chosen location can be those that will be unloaded later than c_{ijk}, that is $l_{i'j'(k'-1)} \in L_{\mathrm{Pod}_{t'}}$ $(t < t')$. Here the larger port index denotes a further port of destination.
- Freeing up space stage: This stage tries to free up an entire row to obtain free space for stack weight adjustment. For example, suppose there are some containers $c_{ijk} \in C_{\mathrm{Pod}_t}$ that exceeded the stack weight limit and needed to be moved out. There are a set of empty locations l' above containers $c_{i'j'k'} \in C_{\mathrm{Pod}_{t'}}$ and $t > t'$. However, in order to avoid the over-stow problem, c_{ijk} is not allowed to be stowed above $c_{i'j'k'}$. In this case, the system will find another stack of containers $c'_{i'j'k'}$ which also belongs to $C_{\mathrm{Pod}_{t'}}$, and move the whole stack of $c'_{i'j'k'}$ to l' without violating the

constraints of stowage. After that, the released space can be used to stow the containers $\mathrm{c_{ijk}}$ that exceeded stack weight limits.

The methods described above are very effective for solving the stack weight problem especially for large number of containers. A case study to illustrate this will be presented in Section 5.

4.2.2 Visibility adjustment

A clear view of sea surface is a pre-condition for safe travelling. Thus, an acceptable stowage plan requires the visibility condition to be satisfied. However, in the basic stowage stage, in order to stow all the containers onto the ship to obtain a preplan, the visibility condition is not considered. After the preplan has been generated, the safety and stability module will adjust the preplan to resolve this issue. The approach is illustrated as follows:

- Swapping step: As container height varies based on its type, the number of containers that can be stowed in one stack is different for different types of containers. For example, one stack may accept four normal containers but only three out-of-gauge containers (OOG) because the height of an OOG is greater than that of a normal one. In this stage, according to the current trim of the ship based on the preplan generated from the stowage plan generator module, the set of containers V that exceeds the visibility limit and are loaded at the current port is identified. Then for each downstream port t, the containers in V that will be discharged at port *t* are grouped into a set $\mathrm{V_{Pod_t}}$. Secondly, we choose a container $\mathrm{c_{ijk}} \in \mathrm{V_{Pod_t}}$ and try to reduce the height of the stack where $\mathrm{c_{ijk}}$ is located in by swap $\mathrm{c_{ijk}}$ with another container $\mathrm{c_{i'j'k'}}$ should satisfy the following: (1) $\mathrm{c_{ijk}} \in \mathrm{C_{Pod_i}}$, (2) $h_{ijk} > h_{i'j'k'}$, and (3) $\sum_{k \geq t_{mini'j'}} h_{i'j'k} + h_{ijk} - h_{i'j'k'} \leq Max_{i'}$.
- Shifting step: If there are still containers remaining in V, these containers should be moved to another empty location. The empty location $l_{i'j'k'}$ which c_{ijk} will be moved to should satisfy two conditions:
  1) The current height of the stack location $l_{i'j'k'}$ belonging to should not exceed the visibility limit. After c_{ijk} is moved to $l_{i'j'k'}$, the total height of the stack should still be less than the limit $Max_{i'}$. So we have $\sum_{k \geq t_{mini'j'}} h_{i'j'k} + h_{ijk} \leq Max_{i'}$.
  2) If the location $l_{i'j'(k'-1)}$ exists, the port of destination of $c_{i'j'(k'-1)}$ should be the same as the container c_{ijk}. If none of the containers that exceeded the visibility limit can be moved to the location below which the container has the same port of destination, then the empty location below which the container will be discharged later is acceptable.
- Freeing up space stage: As the visibility constraint is a hard constraint, we have to find more free space to load the containers that still blocks the line of visibility after the shifting step. This approach is very similar with the one explained in Section 4.2.1. The only difference is that in this stage, the visibility limit will be checked instead of the stack limit.

4.2.3 Trim adjustment (Cross balance)

As shown in Figure 2, containers are stowed in a ship in a bay by bay fashion. Constraint (2) shows that the further distance between the CG of the ship and the bay, the larger will be the longitudinal moment caused by a container in that bay. To illustrate this algorithm, we consider the case $moment_{stern} > moment_{bow.}$ The method for trim adjustment is expressed as follows:

- Step 1: We assume that all containers $\mathrm{c_{ijk}} \in \mathrm{C_{Pod_t}}$ are stowed into more than one bay in the stern side. We choose the heaviest container $\mathrm{c_{ijk}}$ to swap with a lighter container $\mathrm{c_{i'j'k'}}(\mathrm{i} > \mathrm{i'})$ without violating the constraints of ship except constraint (3). After swapping, $\mathrm{moment_{stern}}$ is reduced by

$(w_{c_{ijk}} - w_{c_{i'j'k'}})(x_{ijk} - x_{i'j'k'})$. Similarly, in the bow side of the ship, we choose the lightest container c_{ijk} stowed near the bow to swap with a heavier container $c_{i'j'k'}(i < i')$ stowed near the centre of the ship. The moment_{bow} will be increased correspondingly.

- Step 2: If the difference in the moments between the stern and bow still exceeds the allowable range for the desired trim after Step 1, we choose the container $c_{ijk} \in C_{Pod_t}$ in the stern side to swap with the container $c_{i'j'k'} \in C_{Pod_t}$ in the bow side. The two containers should satisfy the conditions that are listed as follows:

  a) According to the constraint (3), we can obtain the difference in moment for swapping two containers loaded at the stern and bow side respectively as $\mathrm{Mom} = (x_{ijk} + x_{ijk})(w_{c_{ijk}} - w_{c_{ijk}})$ and new stern moment $moment_{stern}' = moment_{stern} - Mom$. Our objective is to reduce stern moment, so Mom should be larger than zero.

  b) However, if $Mom > (moment_{stern} - moment_{bow})$, the case will change to $moment_{stern} < moment_{bow}$. Then we have to repeat from Step 1. So in order to avoid endless loops, we maintain the invariance $\mathrm{Mom} < (\mathrm{moment}_{stern} - \mathrm{moment}_{bow})$.

- Step 3: Containers in the stern side are moved to empty locations in the bow side. The difference in moments between the stern and the bow is reduced by $w_{c_{ijk}}(x_{ijk} + x_{i'j'k'})$.

4.2.4 Heel angle adjustment (Horizontal balance)

To avoid affecting the cross balance, we will adjust the horizontal stability bay by bay. The basic idea is that making each bay balanced horizontally will result in the whole ship being balanced horizontally. In this stage, we assume the transverse moment in bay i is greater than zero. The left side of the ship is heavier than the right side and the difference is Mom_{Tdiff}.

- Step1: In one bay, firstly, we divide all the containers in the same bay into different groups based on their port of destinations and also divide them based on which side of the ship they are located (right side or left side). Then we adjust the horizontal stability port by port. Suppose two containers c_{ijk} and c_{ijk} ($j \in L, j \in R$) which have the same port of destination are stowed at the left side and right side respectively. Assume that the moment on the left side of the ship is larger than that of the right side (the converse is treated in the same manner as follows). Given that the distance between the middle of the bay i and the location l_{ijk} is y_{ijk}, thus the change in transverse moment caused by swapping these two containers is $\mathrm{Mom}_{change} = (y_{ijk} + y_{ijk}) * (w_{ijk} - w_{ijk})$. In order to reduce the left transverse moment and to achieve convergence, containers are only swapped if $\mathrm{Mom}_{change} > 0$ and $\mathrm{Mom}_{change} < \mathrm{Mom}_{Tdiff}$.
- Step 2: After the adjustment by swapping containers, if there is still a difference in moment between the left and right side of the ship, we try to move containers from the left to the right side of the ship or vice versa. The container c_{ijk} selected should be loaded at this port $c_{ijk} \in C_{Pod_t}$. In addition, if the container is moved to location $l_{ij'k'}$, the port of destination of the container loaded at the location below $l_{ij'k'}$ should be greater than $Pod_{c_{ijk}}$ to avoid the over-stow problem. Furthermore, the change in transverse moment due to the moving of this container should be less than Mom_{Tdiff} for convergence. So we have $(y_{ijk} + y_{ij'k'}) * w_{ijk} < \mathrm{Mom}_{Tdiff}$.

4.2.5 Ballast adjustment

As the combination of stowage constraints of large container ship (e.g. container discharge orders, hazarder cargoes limitation) limits the number of available containers and locations for stability adjustment, ballast adjustment may be required to achieve the desired stability requirement for a stowage plan. Currently, planners have to manually adjust the ballast configuration of a stowage plan based on their experience. This process can take up to half an hour for a typical stowage plan.

The objective of automated ballast adjustment in this paper is to achieve an even trim through increasing or reducing ballast volume in different ballast tanks while not affecting the stability conditions achieved for the stowage plan generated in the previous section by using container weights adjustment.

To demonstrate the ballast adjustment algorithm, we consider the case $moment_{stern} > moment_{bow}$ and $w_{Tank_i} = 0,\ i \in T$. The Lcg_i and Tcg_i of $\mathrm{Tank_i}$ will be updated through a lookup table as the volume of the tank (and hence its CG) changes.

- Step1: The twin tanks are linked together (e.g. tanks (2, 3), (4, 5), (7, 8), (9, 10)) as shown in Figure 3. These tanks are then rearranged based on their Lcg in empty state.

- Step2: The objective of ballast adjustment is to achieve $\mathrm{moment_{stern}} = \mathrm{moment_{bow}}$. Therefore, filling in tanks from the bow side to the stern side will reduce ballast.
  a) Firstly, the tanks are rearranged based on the distance between the CG of $Tank_i$ and the longitudinal centre of the ship (Lcg_i).
  b) Secondly, in order to maintain the heel angle, the increased transverse moment by ballast should be even on the left side and the right side of the ship. Before filling in the tank, we find the twin tank for the tank which is not located at the center of the ship. The expected weight will be divided by 2 and loaded in both of them. Thus we have $\left(\frac{\mathrm{w}}{2}\right) * \mathrm{Tcg_i} = \left(\frac{\mathrm{w}}{2}\right) * \mathrm{Tcg_{Twin_i}}$.

- Step3: As the location parameters ($\mathrm{Lcg_i}, \mathrm{Tcg_i}$ etc.) of tank is dynamic following the change of the ballast volume inside, the program should recalculate all the objective values (Trim, GM) after the change. Sometimes, the values may not be the same as the expected value based on the new location information of the tank. To resolve this issue, if $\mathrm{moment_{stern}} - \mathrm{moment_{bow}}$ is much larger (50%) than the maximum longitudinal moment ($\mathrm{w_{Tank_{maxi}}} * \mathrm{Lcg_i}$) created by a special tank, the tank will be fully filled in. Otherwise, we increase the capacity of the tank until $\mathrm{p} * \mathrm{w_{Tank_{maxi}}} * \mathrm{Lcg_{inew}} = \mathrm{moment_{stern}} - \mathrm{moment_{bow}}$.

In conclusion, by using our local search and ballast adjustment algorithm, the issues about the safety and stability of a stowage plan can be solved rather effectively. Thus the stowage plan generated by the safety and stability module is a feasible stowage plan with improved safety and stability.

5 CASE STUDY

In our testing, we consider a real-world containership with a capacity of 7000 TEUs. For convenience, the voyage of the containership is assumed to be H-A-B-C-D-E-F-G-H. Table 1 shows that the number of re-handles required at each port based on the stowage plan generated by our system is very small compared to number of containers unloaded and loaded at the respective ports. Compared to the plans generated by human planners, our plans have much fewer or no re-handles required at the respective ports.

Table 2 shows the number of stacks exceeding stack limit in the different adjustment stages. Our safety and stability adjustment module is able to resolve all the stack weight problems. In fact, in the case of stack weight limit adjustment, we move containers and put them into other bays. Thus there is an impact on the respective crane workload at downstream ports [14]. Table 2 shows that the number of containers moved during stack weight adjustment is small. Thus, any negative impact to crane intensity of downstream ports is not likely to be significant.

As shown in
Table **3**, before visibility adjustment, some containers are loaded at the bow side of the ship and these containers blocked the line of visibility. At this stage, our program tries to solve this problem by exchanging or moving containers. The result displayed in
Table **3** shows that the approach illustrated in Section 4.2.5 is effective for visibility adjustment. However, in port C, as the containers are loaded too high in the basic loading stage, the total height of containers cannot be reduced further by swapping containers. Further adjustments by moving or freeing up space to re-stow the containers allows the visibility problem to be resolved.

Table 4 shows that before the ballast adjustment, for most of the ports, the trim of the ship at each port has been reduced and is close to the desired value. However, in port G, as the number of containers to be loaded is small, there are not enough containers for carrying out stability adjustment containers swapping. At this stage, our stowage plan generator achieved the desired stability requirement through adding extra ballast. While this method may increase the overall load of the ship, it is however effective for trim adjustment without affecting the heel angle as shown in Table 5.

Table 5 shows the result of the adjustment for horizontal balance. As our stowage plan generator module tries to load containers in one bay symmetrically. It means, as shown in Figure 2, that if we stow a container at row 3, we will stow a container which has the same port of destination at row 4. This approach provides more space for heel-angle adjustment. Therefore, from Table 5, we can see that the horizontal balance of stowage plans for most of the ports has been improved after stability adjustment.

Table 1: Number of Re-handle for each port

	A	B	C	D	E	F	G	H
No. of containers unloaded	312	645	0	1086	484	124	0	774
No. of containers loaded	794	564	722	1196	529	448	42	900
No. of re-handles (Stowage PlanGenerator)	0	0	0	0	5	0	2	0
No. of re-handles (Human Planner)	9	11	7	11	51	2	0	14

Table 2**:** Comparison of the Number of Stack Exceeding Stack Limit at Different Stages

		A	B	C	D	E	F	G	H
Stack weight	Preplan	2	69	52	16	2	8	0	0
	After exchanging	0	23	38	2	0	4	0	0
	After moving	0	13	7	0	0	0	0	0
	After freeing up	0	0	0	0	0	0	0	0

Table 3: Comparison of the Number of Container Exceeding Visibility limit at Different Stages

		A	B	C	D	E	F	G	H
Visibility	Preplan	0	47	34	27	2	0	0	0
	After swapping	0	41	34	20	0	0	0	0
	After shifting	0	9	7	3	0	0	0	0
	After freeing up	0	0	0	0	0	0	0	0

Table 4**:** Comparison of Trim (meter) at Different Stages

		A	B	C	D	E	F	G	H
Trim	Preplan	5.85	1.96	1.49	1.86	6.50	2.15	4.61	3.68
	After exchanging	2.64	0	0.99	1.54	2.69	0.49	4.42	3.40
	After moving	0.50	0	0.64	0.67	0	0.35	4.02	2.46
	After Ballast adjustment	0	0	0	0	0	0	0	0

Table 5: Comparison of the Heel-Angle (degree) Before and After Stability Adjustment

		A	B	C	D	E	F	G	H
Heel Angle	Preplan	2.96	0.76	-1.95	-2.43	-0.76	-0.38	1.88	1.04
	After exchanging	0.00	0.01	-0.30	-0.01	0.00	0.00	0.21	0.00
	After Ballast adjustment	0.00	0.01	-0.27	-0.01	0.00	0.00	0.21	0.00

6 CONCLUSION

In this paper, the weight distribution problem of stowing containers into a large containership is discussed. We presented the stability adjustment module which is developed to improve the stability of a stowage plan automatically by a heuristic algorithm. This approach is useful in practice for large containerships. From the results reported, we can see that the weight of containers is distributed reasonably and the stack weight, cross stability and horizontal stability have been improved. Moreover, partial tank information of a ship has been included in our program. Our stability adjustment algorithm has been shown to be effective in solving safety and stability issues for stowage plans. However, currently we have only considered the ballast adjustment for trim and heel angle and did not consider other stability issues such as bending moment and torsion. This will be considered in the next phase of the project. Furthermore, we also plan to develop an optimization engine which will analyse the profile of the containers to be loaded before choosing locations to stow them.

ACKNOWLEDGMENTS

The study is supported by a grant from the Maritime and Port Authority of Singapore and APL. The team acknowledges the contributions of APL colleagues in sharing their knowledge and experience for the stowage project.

REFERENCES

Ambrosino D., and A. Sciomachen. 1989. A constraints satisfaction approach for master bay plans. *In Maritime Engineering and Ports*, eds. G. Sciutto, and C.A. Brebbia, 155-164, WIT Press, Boston.

Ambrosino, D., A. Sciomachen, and E. Tanfani. 2004. Stowing a Containership: The Master Bay Plan problem. *Transportation Research A*, 38(2):81–99.

Aslidis T. 1989. *Combinatorial algorithms for stacking problems*. Ph.D. Thesis, MIT.

Aslidis T. 1990. Minimizing of overstowage in container ship operations. *Operational Research*, 90:457-471.

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

Avriel, M., M. Penn, N. Shpirer, and S.Witteboon. 1998. Stowage Planning for Container Ships to Reduce the Number of Shifts. *Annals of Operations Research*, 76:55–71.

Avriel M., M. Penn and N. Shpirer. 2000. Container ship stowage problem: complexity and connection capabilities. *Discrete Applied Mathematics*, 103:271-279.

Davies A.P., and E.E. Bischoff. 1999. Weight Distribution Considerations in Container Loading. *European Journal of Operational Research*, 114:509–527.

Dubrovsky O., G. Levitin, and M. Penn. 2002. A genetic algorithm with compact solution encoding for the container ship stowage problem. *Journal of Heuristic*, 8:585-599.

Imai A., and T. Miki. 1989. A heuristic algorithm with expected utility for an optimal sequence of loading containers into a containerized ship. *Journal of Japan Institute of Navigation*, 80:117-124.

Wilson I.D., and P.A. Roach. 1999. Principles of combinatorial optimization applied to container-ship stowage planning. *Journal of Heuristics*, 5(4):403-418.

Wilson I.D., and P.A. Roach. 2000. Container stowage planning: a methodology for generating computerised solutions. *Journal of Operational Research Society*, 51(11):1248-1255.

Wilson I.D., P.A. Roach, and J.A. Ware. 2001. Container stowage pre-planning: using search to generate solutions, a case study. *Knowledge-Based Systems,* 14(3-4):137-145.

Xiao X., M.Y.H. Low, F. Liu, S.Y. Huang, W.J. Hsu, and Z.P. Li, 2009. An Efficient Block-Based Heuristic Method for Stowage Planning of Large Containerships with Crane Split Consideration. In *Proceedings of the 2009 International Conference on Harbor, Maritime & Multimodal Logistics. Modelling and Simulation*. Tenerife, Canary Islands, Spain.

AUTHOR BIOGRAPHIES

MIN ZENG is a Research Associate in the Parallel and Distributed Computing Centre, School of Computer Engineering at Nanyang Technological University. Her research interest is in the area of heuristic optimization and simulation. Her email address is <zengmin@ntu.edu.sg>.

MALCOLM YOKE HEAN LOW is an Assistant Professor in the School of Computer Engineering at the Nanyang Technological University (NTU), Singapore. His research interests include parallel computing, modelling and simulation, planning and scheduling optimization. His email address is <yhlow@ntu.edu.sg>.

WEN JING HSU is an Associate Professor with the School of Computer Engineering at the Nanyang Technological University. His research interests include parallel and distributed processing and provably efficient algorithms. His email address is <hsu@ntu.edu.sg>.

SHELL YING HUANG is currently with the School of Computer Engineering, Nanyang Technological University. Her research interests are in intelligent agents, multi-agent systems, simulation based optimization and intelligent decision support systems. Her email address is <assyhuang@ntu.edu.sg>.

FAN LIU is a Project Officer with the School of Computer Engineering at the Nanyang Technological University. His research interests include optimization algorithms and simulation based optimization. His email address is <liufan@ntu.edu.sg>.

CHO AYE WIN is a Project Officer with the School of Computer Engineering at the Nanyang Technological University. His research interests are in the area of computer graphics, high performance visualization, and real time modeling and simulation system. His email address is <wcaye@ntu.edu.sg>.