Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol III,
IMECS 2010, March 17 - 19, 2010, Hong Kong

Improving Ship Stability in Automated Stowage Planning for Large Containerships

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

***Abstract*— Stowage planning for container vessels concerns the core competence of the shipping lines. As such, automated stowage planning has attracted much research in the past two decades, but with few documented successes. In an ongoing project, we are developing a prototype stowage planning system aiming for large containerships. The system consists of three modules: the stowage plan generator, the stability adjustment module, and the optimization engine. This paper mainly focuses on the stability adjustment module. The objective of the stability adjustment module is to check the global ship stability of the stowage plan produced by the stowage plan generator and resolve the stability issues by applying a heuristic algorithm to search for alternative feasible locations for containers that violate some of the stability criteria. We demonstrate that the procedure proposed is capable of solving the stability problems for a large containership with more than 5000 TEUs.**

Keywords— Automation, Stowage Planning, Local Search, Heuristic algorithm, Stability Optimization

I. INTRODUCTION

Stowage planning, or more specifically, the Master Bay Plan Problem (MBPP), is formally described in (Ambrosino and Sciomachen, 2004). It is a difficult problem 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 [1] or the number of shifts [2] rather than weight distribution.

Imbalance in weight distribution of containers onboard a ship can cause ship stability problems and lead to disasters. Currently, the weight distribution of containers in a stowage plan is still carried out manually by human planners based on their experience. With the capacity of the bigger containerships reaching ten thousand TEUs (Twenty Foot Equivalent Unit) and more, it is challenging to manually generate a stowage plan with a good weight balance.

The objective of our study is to develop a fully automated system for stowage planning for 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 ship stability consideration; (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 [14], in this paper, we only present the related work on the stability adjustment. 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.

II. LITERATURE REVIEW

Since the 1970s, the problem related to container stowage planning 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 [4] [13]. 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 an optimal solution in a reasonable processing time [1]. Meanwhile, several researchers try to develop 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 was 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 including the expected number of containers to shift, and 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.

Manuscript received January 12, 2010. This work was supported by the Maritime and Port Authority of Singapore and American President Lines.

Zeng Min, Malcolm Yoke Hean Low, Hsu Wen Jing, Huang Shell Ying, Liu Fan, Win Cho Aye are with the Parallel and Distributed Computing Centre, School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798 (phone: 65-65132178; e-mail: {zengmin, yhlow, hsu, assyhuang, liufan, wcaye}@ntu.edu.sg).

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

However, in their approach, they only considered one metric, viz., *GM*, in the ship stability issue. Other factors such as *Heel Angle* and *trim* were ignored. This assumption makes their approach inapplicable to solving real stowage problems.

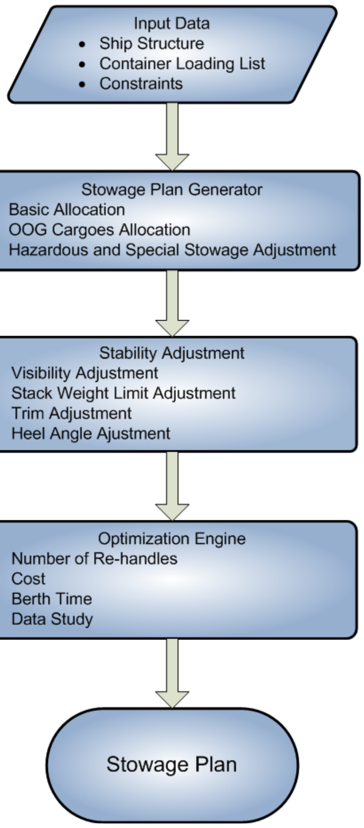

Figure 1. System of automated stowage planning

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. Averiel 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 only has a large cargo bay, and did not consider the issues of hatch covers and stability. Also, Averiel et al. (2000) showed that the stowage planning problem is NP-complete by showing that the stowage problem is related with a known NP-hard problem, viz. the circle graphs coloring problem.

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 using a genetic algorithm approach. However, the approach to generate a stowage plan still needs 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, his work only considered the ship to have a small, single bay, and he also ignored the stability issue which is very critical in stowage planning.

In the papers of Ambrosino et al. (1998, 2004, 2006), the stowage planning problem is called the Master Bay Plan Problem (MBPP). Ambrosino and Sciomachen (1998) reported the first attempt to derive the constraints [7] 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 MBPP by introducing a tolerance of move count from the perspective of cranes. By setting the tolerance to a suitable value, the algorithm can generate a stowage plan with less number of re-handles and efficient utilization of cranes, which are two important objectives of MBPP. They tried to deal with the ship stability issue as well and mentioned an approach which supposedly solves the stack weight problem. However, for large containerships with many heavy containers, their approach is unlikely to resolve all the problems.

Since all the research mentioned above were carried out under simplistic assumptions (except for the work of Xiao et. al. (2009)), and since existing results seldom consider the stability problem, they can hardly 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 all the existing containership features and constraints to rapidly generate a set of feasible plans for a containership on a multi-port voyage. A feasible plan is one that is safe for the ship to sail.

III. THE STRUCTURE OF A CONTAINERSHIP

The stowage planning problem is to assign a given set C of n containers with different properties to a set L of m available locations of a containership. The cross section of a typical containership is shown in Figure 2. A containership contains a number of bays with ID numbers increased from bow to stern. There are two types of bays. A 40 foot (40') bay is counted in even number and reserved for 40' containers. A 20 foot (20') bay is counted in odd number and reserved for 20' containers. However, two adjacent 20' bays can be used as one 40' bay, such as bay14 = bay15+ bay13. Each bay includes rows that are numbered from centre to outside and tiers, numbered from bottom to top. We present definitions as follows:

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

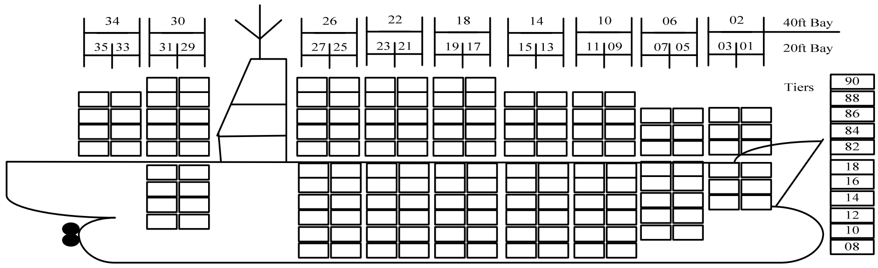

Figure 2. Cross sections of a containership

- l_{ijk} is defined as a location in a bay with 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 tier number in bay i, row j under deck.
- t_{minij} is defined as the minimum tier number in bay i ,row j at on deck.
- x_{ijk} is defined as the distance between the centre of gravity of the ship and the location l_{ijk}.
- y_{ijk} is defined as the distance between the centre of gravity of bay i and the location l_{ijk} in the same bay.
- c_{ijk} is defined as a container which is stowed at the bay i, row j and tier k.

In addition, the properties of containers also affect the stowage planning process. 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.
- Port of destination: As we know, containers have their own destinations. If the containers that will go to a further port are loaded above the containers that will be unloaded first, we call this case as over-stow. Over-stow will cause re-handling operations in the subsequent port. In order to minimize unnecessary re-handling and re-shuffling of containers during unloading at a port, containers going to further ports are loaded first and the containers which will be unloaded first should be loaded last. We define Pod_c as the destination of container c, C_{Pod_t} as a set of containers which are loaded at 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}.
- Weight: Five categories of weight 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. d is 0 if the container is under deck or 1 if it is on deck. The issue of weight distribution will be further illustrated in Section 4.

IV. STABILITY IMPROVEMENT ALGORITHM

As shown in Figure 1, the stability module includes four procedures aimed at optimizing visibility, stack weight, trim and heel-angle. The purpose of visibility adjustment is to make sure the view of the sea surface from the navigation bridge is not blocked. In this paper, we focus on studying the effects on ship structure caused by the weight distribution of containers. Thus the details of visibility adjustment will be ignored and discussed in our future work. The stability conditions and algorithms about the stack weight limit, *trim* (i.e., the moment balance between bow and stern) and *heel angle* (i.e., the moment balance between the left side and the right side) are presented as follows.

A. *The stability condition.*

The stowage plan generated by our Stowage Plan Generator module is based on different sizes, types and destinations of containers. However, the weights of containers are not considered in the stowage plan generation. Therefore, we develop the stability module to check weight distribution and produce a feasible stowage plan that satisfies the stability conditions specified below.

$$\sum w_{c_{ijk}} \le w_{maxijd} \text{ (If d=0, k}\le t_{maxij}\text{, k}\ge t_{minij}\text{)} \quad (1)$$

$$-M_1 \le \sum_{i\in A,i,k} w_{c_{ijk}} x_{ijk} - \sum_{i\in F,i,k} w_{c_{ijk}} x_{ijk} \le M_1 \ (2)$$

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

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) supposed to be within 0.5 meter, where the $Trim$ is defined as the difference of 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 change in the trim of one centimeter.

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 constraints (2), where A is a set of bays on the stern and F is a set of bays on the bow.

Finally, the horizontal stability condition requires that the moment on the right side and that of the left side of a ship must not differ by more than a given tolerance M_2. This

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

condition is expressed by constraints (3), where L is a set of rows in left side of a ship and R is a set of rows in right side of a ship.

B. Adjustment algorithm

For the stability adjustment, we develop a simple yet effective heuristic algorithm. We first adjust the stack weight of containers. Then, if needed, adjustments are carried out to balance the moments between stern and bow. Finally, we adjust the weight of containers in the same bay to ensure the balance of moments between the right and left side of the ship. The adjustments in each step are carried out such that the balance effects of the previous steps are not affected.

1) Stack weight adjustment:

The stack weight limit condition is important for stowage planning. If the weight of the containers in one stack exceeds the stack weight limit, the stack may collapse during voyage. We take three steps to deal with this problem.

- Exchange stage: Firstly, to get 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 limit. Then we choose the heaviest container $c_{ijk} \in C_{Pod_t}$ in one of the stack in S_{exceed} to swap with a lighter container $c_{i'j'k'} \in C_{Pod_t}$ of the stack in S_{free}.

- Moving stage: If there is not enough feasible containers to swap with in the Exchange stage, in order to reduce the stack weight, the container $c_{ijk} \in C_{Pod_t}$ at the top of one of the stack in S_{exceed} are moved to some other locations subject to certain constraints, as detailed below:

  - Firstly, choose the empty location which satisfies three conditions:

    (a) The empty location $l_{i'j'k'}$ should be located at the stack which 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., $w_{c_{ijk}} + \sum_k w_{c_{i'j'k'}} \le w_{maxi'j'd}$(if d=0, k$\le t_{maxij}$, k$\ge t_{minij}$).

    (c) The locations below of the chosen location should have been filled up with containers of the same port of destination with the container c_{ijk}.

  - Secondly, choose the location $l_{i'j'k'}$ which satisfies the first two conditions (a), (b) mentioned above. And the below container can be the one will be unloaded later than c_{ijk}, that is $l_{i'j'(k'-2)} \in L_{Pod_{t'}}$ $(t < t')$. 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 $c_{ijk} \in C_{Pod_t}$ that exceeded the stack weight limit and needed to be moved out. There are a set of locations l' above containers $c_{i'j'k'} \in C_{Pod_{t'}}$ and $t > t'$ are empty. However, in order to avoid over-stow problem, c_{ijk} is not allowed to be stowed above $c_{i'j'k'}$. The system will find a stack of containers which also belong to $C_{Pod_{t'}}$, and move the whole stack to l' without violating the constraints of stowage. After that, the released space can be used to stow the containers 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.

2) 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 the distance between the centre of gravity of the ship and the bay, the larger will be the longitudinal moment caused by a container in that bay. To illustrate the working of the 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 $c_{ijk} \in C_{Pod_t}$ are stowed into more than one bay in stern side. We choose the heaviest container c_{ijk} to swap with a lighter container $c_{i'j'k'} (i > i')$ without violating the constraints of ship except constraint (3). After swapping, $moment_{stern}$ 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 stowed near the bow to swap with a heavier container stowed near the centre. The $moment_{bow}$ will be increased correspondingly.
- Step 2: If the difference in the moments between the stern and bow still exceeds the tolerance 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 get the difference of moment for swapping two containers loaded at the stern and bow side respectively as $Mom = (x_{ijk} + x_{i'j'k'})(w_{c_{ijk}} - w_{c_{i'j'k'}})$ 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

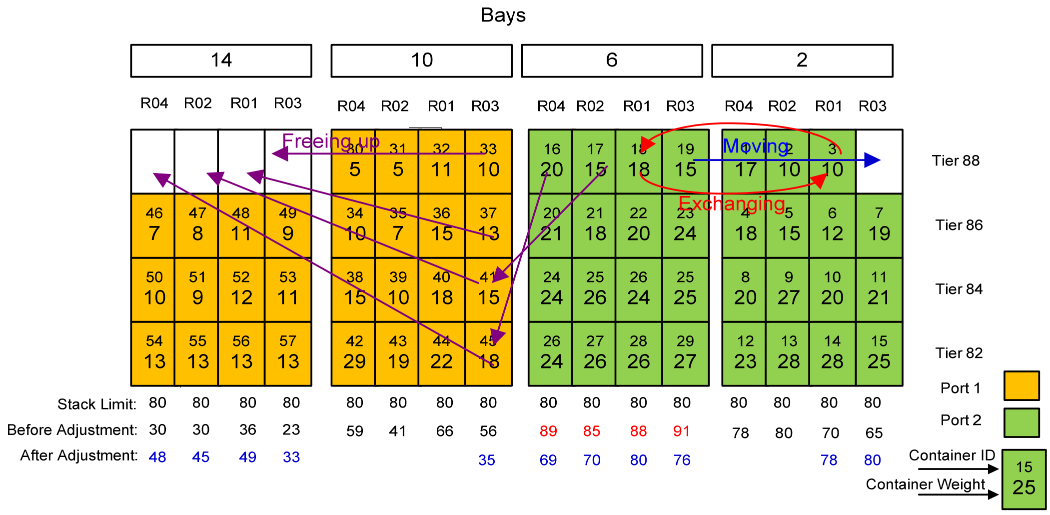

Figure 3. Stack weight adjustment

we have to repeat from the step 1. So in order to avoid endless loops, we maintain the invariance condition $Mom < (moment_{stern} - moment_{bow})$.

- Step 3: to move the containers in the stern side to an empty location in the bow side. The difference of moments between the stern and the bow is reduced by $w_{c_{ijk}}(x_{ijk} + x_{i'j'k'})$.

3) 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 not zero. The left side 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 to different groups based on their port of destinations. Then we deal with the horizontal stability port by port. We have two containers c_{ijk} and $c_{ij'k'}$ $(j \in L, j' \in R)$ which belong to the same group. By assumption, the moment on the left side is larger than that of the right side. The change of transverse moment caused by swapping these two containers is $Mom_{change} = (y_{ijk} + y_{ij'k'}) * (w_{ijk} - w_{ij'k'})$. In order to reduce the left transverse moment, we have $Mom_{change} > 0$. Also in order to avoid endless loop, we limit $Mom_{change} < Mom_{Tdiff}$.
- Step 2: After adjustment by swapping containers, if there is still a difference in moment between the left and right side, we try to move the container loaded at the left side to the right side. Firstly, the container c_{ijk} 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 of moments by moving should be less than Mom_{Tdiff} to avoid endless loop, so we have $(y_{ijk} + y_{ij'k'}) * w_{ijk} < Mom_{Tdiff}$.

In this stage, we can swap two containers in the same bay.

In conclusion, by using our local search algorithm, the weight distribution problems of a stowage plan have been solved rather effectively. Thus the stowage plan generated by the stability module is a feasible stowage plan with improved stability.

V. CASE STUDY

In order to illustrate how the stability adjustment algorithm is carried out, we present a case study for stack weight adjustment in this section. We assume that there are three ports, the current port is Port 0, and the sequence of voyage is $Port\ 0 \rightarrow Port\ 1 \rightarrow port\ 2$. As shown in Figure 3, there are four 40' bays. The containers in yellow that will go to $Port\ 1$ are stowed in bays 10 and 14 and the containers in green that will go to $Port\ 2$ are stowed in bays 2 and 6. There are four rows (R01-R04) in bay 6 which exceed the stack weight limit and the total weight is denoted in red numbers in Figure 3. We assume that all containers are loaded at $Port\ 0$.

Firstly, we find that if c_{060188} swaps with c_{020188}, the stack problem in bay 6 and row 01 can be solved. Secondly, as there are still some stack weight problems that cannot be solved by exchanging two containers, we try to move containers to empty locations. Although there are some empty locations in bay 14, the yellow containers (container 30-57) will be unloaded first. Therefore, these empty locations are not available for green containers (container 1-29). There is only one empty location in bay2 that can be used. We move the container $c_{060388}$to location l_{020388}. The weight of stack in bay 06 and row 03 reduces to 78 tons which is less than the stack limit of 80 tons.

After the Exchanging and Moving stages, we still have two stacks that exceed stack limit. In order to solve the stack weight problems, we have to free up more space. As the yellow containers will be unloaded first, we avoid putting the green one on the yellow one directly. However, we can free up row 03 in bay 10 by moving the four yellow containers in row 03 from bay 10 to bay 14. Then the stack weight problems are solved completely.

In our testing, we consider a containership with a capacity of 5000 TEUs. The voyage of the containership is given as H-A-B-C-D-E-F-G-H. Table 1 shows the number of stacks exceeding stack limit in the different adjustment

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

stages. Our 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 effect on their loading/unloading time. However, as shown in Table 1, the number of containers moved in stack weight adjustment is small, so the negative effect for crane split [14] is not significant.

Table 2 shows that 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 containers swapping. In the next stage of our work, we plan to carry out automated exchange of a whole bay in order to improve the stability adjustment algorithm.

Table 3 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 03, we will stow a container which has the same port of destination at row 04. This approach provides more space for heel-angle adjustment. Therefore, from Table 3, we can see that most of horizontal balance problems have been improved after stability adjustment.

VI. 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. However, currently we have not yet included the tank information of a ship (e.g. fuel tanks, ballast tanks) in our stowage plan. 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.

ACKNOWLEDGMENT

The study is supported by grants from the Maritime and Port Authority of Singapore and APL.

REFERENCES

[1] Ambrosino, D., A. Sciomachen, and E. Tanfani. (2004). “Stowing a Containership: The Master Bay Plan problem.” Transportation Research A 38, 81–99.
[2] 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.
[3] Davies, A.P., and E.E. Bischoff. (1999).“Weight Distribution Considerations in Container Loading.” European Journal of Operational Research 114, 509–527.
[4] Aslidis T. Combinatorial algorithms for stacking problems, Ph.D. Thesis, MIT, 1989.
[5] Aslidis T. Minimizing of overstowage in container ship operations, Operational Research, 1990, Vol. 90, pp. 457-471.
[6] Avriel M. and Penn M. Exact and approximate solutions of the container ship stowage problem, Computers and Industrial Engineering, 1993, Vol. 25, pp. 271-274.
[7] Ambrosino D and Sciomachen A. A constraints satisfaction approach for master bay plans, In: Sciutto, G., Brebbia, C.A. (Eds.), Maritime Engineering and Ports, 1989, pp. 155-164, WIT Press, Boston.
[8] Avriel M, Penn M and Shpirer N. Container ship stowage problem: complexity and connection capabilities, Discrete Applied Mathematics, 2000, Vol. 103, pp. 271-279.
[9] Dubrovsky O, Levitin G and Penn M. A genetic algorithm with compact solution encoding for the container ship stowage problem, Journal of Heuristics, 2002, Vol. 8, pp. 585-599.
[10] Imai A and Miki T. A heuristic algorithm with expected utility for an optimal sequence of loading containers into a containerized ship. Journal of Japan Institute of Navigation, 1989, Vol. 80, pp. 117-124.
[11] Wilson ID and Roach PA. Principles of combinatorial optimization applied to container-ship stowage planning, Journal of Heuristics, 1999, Vol. 5, pp. 403-418.
[12] Wilson ID and Roach PA. Container stowage planning: a methodology for generating computerised solutions, Journal of Operational Research Society, 2000, Vol. 51, pp. 1248-1255.
[13] Wilson ID, Roach PA and Ware JA. Container stowage pre-planning: using search to generate solutions, a case study, Knowledge-Based Systems, 2001, Vol. 14, pp. 137-145.
[14] Xiao X., Low M. Y. H., Liu F., Huang S. Y., Hsu W. J., Li Z., 2009. The Proceedings of the International Conference on Harbour, Maritime & Multimodal Logistics Modelling and Simulation, pp. 93-99.

Table. 1. 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. 2. Comparison of the number trims at different stages (m)

		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

Table. 3. Comparison of the heel-angle before and after adjustment (angle of heel °)

		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

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)