

# Modified Integer Model for Solving the Master Bay Problem

M. Tsymbal & K. Kamieniev

*National University "Odesa Maritime Academy", Odessa, Ukraine*

**ABSTRACT:** One of key components in keeping a ship seaworthy is a correctly prepared cargo plan. Considering the recent cost cutting measures and reduced transportation volumes relevance of optimizing such a plan cannot be understated. Though there's a number of studies addressing the issue none of them covers all the operational and constructional constraints necessary to factor for. This article presents an integer model that tries to address some of the constraints missed by other researches. A method for solving the model is designed and developed based on a steady-state genetic algorithm. A numerical experiment is conducted showing the method's feasibility.

## 1 INTRODUCTION

Transportation safety is one of the most important problems in shipping. The problem consists of various tasks one of which is ensuring that ships carrying cargoes are kept seaworthy at any point of the route.

Container shipping has historically had razor thin margins and the current pandemic has also cut shipping volumes as well. According to UNCTAD review of maritime transport, 2020, shipping companies started to significantly reduce capacity in the second quarter of 2020 in order to reduce costs and keep freight rates from declining [24]. At the same time the size of the largest container vessel in terms of capacity went up by 10.9% and the global shipping fleet has increased by 4.1% which is the highest since 2014. Cost cutting measures make the problem of keeping ships seaworthy even more relevant.

One of key components in keeping a ship seaworthy is a correctly prepared cargo plan. Considering the abovementioned factors, relevance of

optimizing such a plan is evident. Container ship stowage planning problem is known as MBPP (Master Bay Plan Problem) and its detailed description can be found in [5]. In general, it consists of placing a set of  $n$  containers into a set of  $m$  available slots on the ship while accounting for structural and operational constraints of containers and the vessel itself.

There have been many attempts to solve the problem in the operations research literature. The original work includes a simplified integer programming model solved by dividing the task in 3 parts. The first part consists of excluding container positions that don't satisfy some strict requirements. Such as excluding reefer containers positions that aren't close by electrical sockets from the total set of solutions. The second part separates containers in accordance with their destination ports into different bays; the containers destined to closest ports are stowed closed to the ship's center. The resulting subset is passed to the third part of the algorithm. The third part solves a system of equations incorporating the subset from the second part. The model doesn't

consider stability or strength parameters; authors circumvent this by using a set of simplifications, such as lighter containers are loaded on top of heavier ones, the total weight of containers stowed in the forward part of the vessel is equal (within a margin of error) to the total weight of the ones stowed in the aft etc.

In [18] 3D-BPP (3D Bay Plan Problem) approach is used, considering ship a 3D bin and dividing the containers in a way that allows for parallel discharging [1]. Uses an integer model that separates containers in 3 categories based on their weight (namely light, medium and heavy) and tries to minimize the loading time using tabu search. In [2] the integer model is being solved via a simple heuristics and the ant colony optimization algorithm. [4] uses the 3D-BPP approach and includes certain dangerous cargoes. In [3] the authors propose two additional integer models and heuristic algorithms.

Authors of [8] use a simplified containership model that consists of R rows and C columns and carries cargoes to multiple ports. A genetic algorithm is used trying to minimize container re-handles.

In [20] the problem is solved using a branch and bound method in order to minimize cargo hatch and port cranes movements as well as container re-handles.

Authors of [22] propose a ship model including bays and ballast tanks and a heuristics that shifts containers not satisfying requirements to empty slots.

In [12] authors propose a method of stowing dangerous goods via selecting allowable slows for containers and stowing them randomly in one of those.

Authors of [23] propose four integer models for different loading cases and use third party solvers for obtaining solutions.

In [6] a vessel is simplified to a single  $m \times n$  bay. Authors use a greedy algorithm to minimize containers re-handles.

Authors of [17] view a vessel as a single bay as well and propose an integer model and a heuristics for solving it.

In [21] authors use steepest ascent hill climbing, genetic, and simulated annealing algorithms for solving the problem while viewing a vessel as a single bay.

Authors of [15] use the model from [5] and particle swarm optimization as a method for solving it.

In [9] a genetic algorithm is used for optimizing a solution based on re-handles number, trim and rolling period.

Authors of [19] use a deep reinforcement learning network for optimizing containers re-handles and shore crane movements.

In [16] authors propose a Boolean model and use a GRASP (greedy randomized adaptive search procedure) algorithm to solve the problem.

Authors of [11] propose a two-step heuristics using the divide and conquer paradigm. For obtaining a

solution the strictest constraints are shifted towards the beginning of constraints list which allows to disregard incorrect solutions faster.

In [7] a constraint and an integer programming models are proposed for solving the problem.

Finally, authors of [13] use a tabu search algorithm multi-objective optimization which tries to minimize the number of re-handles, empty container slots, horizontal and longitudinal moments and a single crane discharging time.

The models and methods mentioned above adopt various simplifying assumptions that make them applicable to a limited number of cases. The assumptions include viewing all containers as TEUs (twenty equivalent units), viewing a ship as a single bay, disregarding structural requirements such as impossibility of stowing TEUs on top of FEUs (forty equivalent units) and so on. Because of these the task of constructing a mathematical model that would consider as many constraints as necessary remains unresolved and requires a more detailed study.

## 2 MATHEMATICAL MODEL

This study is a continuation of [10]; it tries to improve the model presented there and solve it using another approach. A container of size  $t$  ( $0=TEU$ ,  $1=FEU$ ), IMDG class  $c$  ( $c=0$  means there's no dangerous goods), is stowed in position  $(i, j, k)$  ( $i$  – bay number,  $j$  &  $k$  are coordinates inside the bay, ) if  $x_{t,c,i,j,k}=1$ .

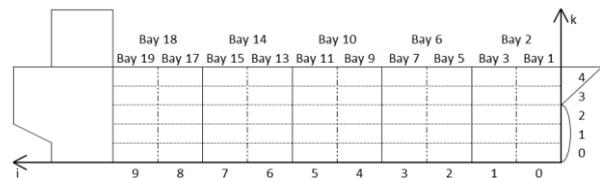


Figure 1. Coordinates  $i$  &  $k$

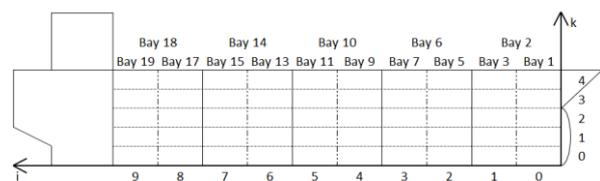


Figure 2. Coordinates  $j$  &  $k$

The following constraints are taken into account:

$$-\left(\sum_i \sum_j \sum_k x_{t,c,i,j,k}\right) \leq (-1-t)n_t^c, \quad \forall t, c, i, j, k; \quad (1)$$

$$\left|x_{1c,i',j,k} - x_{1c,(i'+1),j,k}\right| \leq 0, \quad \forall c, i', j, k; \quad (2)$$

$$\sum_t \sum_c (x_{t,c,i,j,k}) \leq 1, \quad \forall t, c, i, j, k; \quad (3)$$

$$\sum_c \sum_{k^* > k} (x_{0c,i,j,k^*} + x_{0c,(i+1),j,k^*}) + 2M \sum_c x_{1c,i,j,k} \leq 2M, \quad \forall c, i, j, k; \quad (4)$$

$$\left| \left[ \sum_c \sum_{k=0}^k x_{0cijk}^* \right]_s - \left[ \sum_c \sum_{k=0}^k x_{0c(i+1)jk}^* \right]_s \right| \leq M(1-w_s), \quad \forall c, i', j, k; \quad (5)$$

$$\left[ \sum_c x_{tcij(k+1)} + \sum_c x_{tc(i+1)j(k+1)} \right]_s \leq Mw_s, \quad \forall c, i', j, k;$$

$$\sum_{i^*=i'-l_{c1c2}}^{i'+l_{c1c2}} \sum_{j^*=j'-w_{c1c2}}^{j'+w_{c1c2}} \sum_{k^*=k'-h_{c1c2}}^{k'+h_{c1c2}} (x_{t,c1,i^*j^*k^*}) \leq R(1-x_{tc2ijk}),$$

$$\forall t, c1 \neq c2, i'', j'', k''. \quad (6)$$

$$k^* \sum_t \sum_c (x_{tcijk}) - \sum_{k^*=0}^{k^*-1} \sum_t \sum_c (x_{tcijk^*}) \leq 0 \quad \forall t, c, i, j, k > 0; \quad (7)$$

Constants, sets and variables in the model:

$i \in [0..i_{max}]$ , where  $i_{max} + 1$  is the total number of TEU bays;

$j \in [0..j_{max}]$ , where  $j_{max} + 1$  is the maximum width of bays in TEU;

$k \in [0..k_{max}]$ , where  $k_{max} + 1$  is the maximum height of bays in TEU;

$n_c^c$  is the total number of IMDG class  $c$  containers;

$i'$  are FEU bays' numbers;

$M$  is the maximum stack height in TEU;

$w_s \in \{0,1\}$  is an additional variable that is introduced for every  $s$ -th limitation,

$s = i^*(j_{max} + 1) * (k_{max} + 1) + j^*(k_{max} + 1) + k$ ;

$i'' \in [l_{c1c2}..i_{max} - l_{c1c2}]$ , where  $l_{c1c2}$  is a longitudinal separation requirement, in TEU, between two containers of IMDG classes  $c1$  and  $c2$ ;

$j'' \in [w_{c1c2}..j_{max} - w_{c1c2}]$ , where  $w_{c1c2}$  is a transversal separation requirement, in TEU, between two containers of IMDG classes  $c1$  and  $c2$ ;

$k'' \in [h_{c1c2}..k_{max} - h_{c1c2}]$ , where  $h_{c1c2}$  is a height separation requirement, in TEU, between two containers of IMDG classes  $c1$  and  $c2$ ;

Inequality (1) limits the total number of containers to be stowed, (2) makes sure that every FEU container is stowed in two TEU positions, (3) limits the number of containers occupying the same position to one, (4) ensures that no TEUs can be loaded on top of FEUs, (5) checks that no FEU can be loaded on top of TEU stacks of different heights, (6) checks for IMDG code segregation requirements and (7) ensures that all the containers are loaded either on top of each other or on deck.

The model presented does not account for strength or stability requirements, such checks are planned to be performed in the next stage of the study.

### 3 SOLVING THE MODEL

In order to try and solve the abovementioned model a genetic algorithm was selected. Multiple authors used variations of this approach [8, 9, 21] and showed its feasibility. The Genetic Algorithm (GA), often referred to as genetic algorithms, was invented by John Holland at the University of Michigan in the 1970s [14]. It imitates the process of natural selection and is used to generate solutions based on mutation, crossover and selection operations. The classic algorithm requires an initial population of solutions which are received either randomly or via a certain heuristic. After that two parents are selected from the population, crossed over with one another and the result is mutated. This forms children which are then added to the child population. The operation is repeated until the child population is entirely filled. Children's fitness is assessed and the algorithm stops

when it has reached a certain value or when a pre-determined amount of time has passed.

There's also an alternative approach called a steady-state approach which is used in the current study. Its main difference consists of updating the initial population instead of replacing it altogether. Therefore, the children obtained in crossover are reintroduced directly into the initial population removing some preexisting individuals.

The fitness function used creates a Boolean coefficients matrix of the model based on the given containers set. After that it converts the solution to a Boolean variables matrix and multiplies the two. The resulting matrix is compared to the inequalities' constant matrix. Every equation not satisfied increases the candidate's unfitness by 1. Therefore, the higher the result the lower the individual's fitness is.

The mutation function changes individual containers' coordinates within the allowed range [0..imax], [0..jmax] and [0..kmax] not affecting other parameters.

In order to avoid the linkage problem the uniform crossover function is chosen for the task. Similar to the mutation function it crosses over actual containers' coordinates not affecting other parts of the matrix.

For selection a Tournament Selection algorithm is chosen, which picks  $n$  individuals from the initial population and chooses the fittest of those.

The testing is performed using a simplified ship model consisting of 4 TEU bays without covers and 25 closed containers to be loaded.

The initial solution is formed via simply filling the 1st bay (Fig. 3).

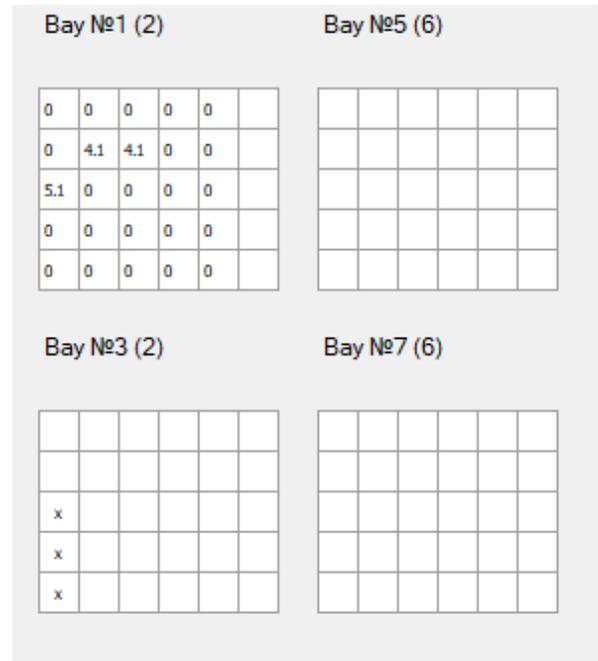


Figure 3. Initial solution

The numbers in cells represent IMDG classes, 0 means the cargo is not dangerous, x is used to indicate a FEU taking up two places.

As we can see two of the requirements are not satisfied, specifically IMDG cargoes segregation, which should be at least one vertical column between the incompatible containers, and TEUs on top of FEUs placement restriction.

The resulting solution (Fig. 4) is obtained by the genetic algorithm and it satisfies all the constraints set in the model. Therefore the model in its current state can be solved using a steady-state genetic algorithm.

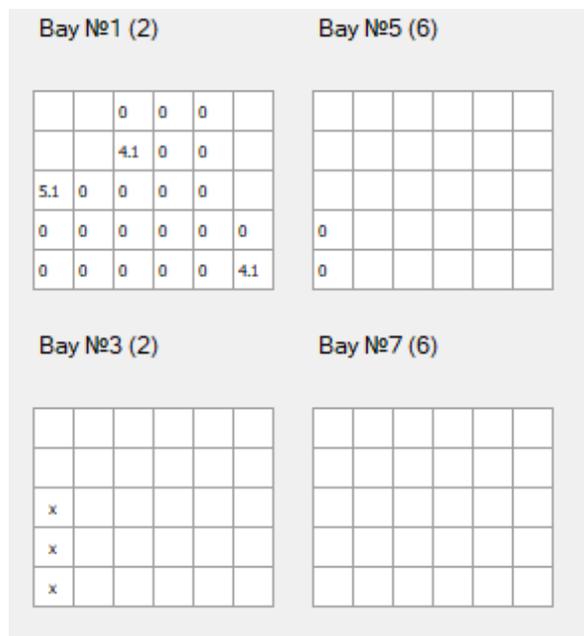


Figure 4. Resulting solution

#### 4 CONCLUSIONS

In this paper previously developed mathematical model for solving the MBPP problem has been modified and presented in a more concise and practical way. A generic steady-state genetic algorithm has been used and its functions have been modified in order to solve the new model taking into consideration the new constraints. A numerical experiment has been conducted and has shown that the developed method can be used to solve the model in its current state.

#### REFERENCES

1. Ambrosino, D., Anghinolfi, D., Paolucci, M., Sciomachen, A.: A new three-step heuristic for the Master Bay Plan Problem. *Maritime Economics & Logistics*. 11, 1, 98–120 (2009). <https://doi.org/10.1057/mel.2008.19>.
2. Ambrosino, D., Anghinolfi, D., Paolucci, M., Sciomachen, A.: An Experimental Comparison of Different Heuristics for the Master Bay Plan Problem. In: Festa, P. (ed.) *Experimental Algorithms*. pp. 314–325 Springer Berlin Heidelberg, Berlin, Heidelberg (2010).
3. Ambrosino, D., Paolucci, M., Sciomachen, A.: Experimental evaluation of mixed integer programming models for the multi-port master bay plan problem. *Flexible Services and Manufacturing Journal*. 27, 2, 263–284 (2015). <https://doi.org/10.1007/s10696-013-9185-4>.

4. Ambrosino, D., Sciomachen, A.: Using a Bin Packing Approach for Stowing Hazardous Containers into Containerships. In: Fasano, G. and Pintér, J.D. (eds.) *Optimized Packings with Applications*. pp. 1–18 Springer International Publishing, Cham (2015). [https://doi.org/10.1007/978-3-319-18899-7\\_1](https://doi.org/10.1007/978-3-319-18899-7_1).
5. Ambrosino, D., Sciomachen, A., Tanfani, E.: Stowing a containership: the master bay plan problem. *Transportation Research Part A: Policy and Practice*. 38, 2, 81–99 (2004). <https://doi.org/10.1016/j.tra.2003.09.002>.
6. D. M. Rahsed, M. S. Gheith, A. B. Eltawil: A Rule-based Greedy Algorithm to Solve Stowage Planning Problem. In: 2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). pp. 437–441 (2018). <https://doi.org/10.1109/IEEM.2018.8607517>.
7. Delgado, A., Jensen, R.M., Janstrup, K., Rose, T.H., Andersen, K.H.: A Constraint Programming model for fast optimal stowage of container vessel bays. *European Journal of Operational Research*. 220, 1, 251–261 (2012). <https://doi.org/10.1016/j.ejor.2012.01.028>.
8. Dubrovsky, O., Levitin, G., Penn, M.: A Genetic Algorithm with a Compact Solution Encoding for the Container Ship Stowage Problem. *Journal of Heuristics*. 8, 6, 585–599 (2002). <https://doi.org/10.1023/A:1020373709350>.
9. Hu, M., Cai, W.: Multi-objective optimization based on improved genetic algorithm for containership stowage on full route. In: 2017 4th International Conference on Industrial Engineering and Applications (ICIEA). pp. 224–228 (2017). <https://doi.org/10.1109/IEA.2017.7939211>.
10. Kamieniev, K., Kamienieva, A., Tsybmal, M.: Construction of a mathematical model and a method for arranging hazardous cargoes on a containership. *EEJET*. 6, 3 (102), 20–27 (2019). <https://doi.org/10.15587/1729-4061.2019.183385>.
11. Lee, Z.Q., Fan, R., Hsu, W.-J.: Optimizing Constraint Test Ordering for Efficient Automated Stowage Planning. In: Corman, F., Voß, S., and Negenborn, R.R. (eds.) *Computational Logistics*. pp. 343–357 Springer International Publishing, Cham (2015).
12. Lei, H., Ok, M.: Dangerous Goods Container Allocation in Ship Stowage Planning. In: *Proceedings of the 9th International Conference on Operations Research and Enterprise Systems - ICORES*. pp. 241–246 SciTePress (2020). <https://doi.org/10.5220/0009160602410246>.
13. Liu, F., Low, M.Y.H., Hsu, W.J., Huang, S.Y., Zeng, M., Win, C.A.: Randomized Algorithm with Tabu Search for Multi-Objective Optimization of Large Containership Stowage Plans. In: Böse, J.W., Hu, H., Jahn, C., Shi, X., Stahlbock, R., and Voß, S. (eds.) *Computational Logistics*. pp. 256–272 Springer Berlin Heidelberg, Berlin, Heidelberg (2011).
14. Luke, S.: *Essentials of Metaheuristics*. lulu.com; Second Edition (2013).
15. Matsaini, Santosa, B.: Solving the Container Stowage Problem (CSP) using Particle Swarm Optimization (PSO). *IOP Conf. Ser.: Mater. Sci. Eng.* 337, 012002 (2018). <https://doi.org/10.1088/1757-899X/337/1/012002>.
16. Parreño, F., Pacino, D., Alvarez-Valdes, R.: A GRASP algorithm for the container stowage slot planning problem. *Transportation Research Part E: Logistics and Transportation Review*. 94, 141–157 (2016). <https://doi.org/10.1016/j.tre.2016.07.011>.
17. Parreño-Torres, C., Alvarez-Valdes, R., Parreño, F.: Solution Strategies for a Multiport Container Ship Stowage Problem. *Mathematical Problems in Engineering*. 2019, 9029267 (2019). <https://doi.org/10.1155/2019/9029267>.
18. Sciomachen, A., Tanfani, E.: A 3D-BPP approach for optimising stowage plans and terminal productivity. *European Journal of Operational Research*. 183, 3, 1433–1446 (2007). <https://doi.org/10.1016/j.ejor.2005.11.067>.

19. Shen, Y., Zhao, N., Xia, M., Du, X.: A Deep Q-Learning Network for Ship Stowage Planning Problem. *Polish Maritime Research*. 24, s3, 102–109 (2017). <https://doi.org/10.1515/pomr-2017-0111>.
20. Wilson, I.D., Roach, P.A., Ware, J.A.: Container Stowage Pre-Planning: Using Search to Generate Solutions, a Case Study. *Know.-Based Syst.* 14, 3–4, 137–145 (2001). [https://doi.org/10.1016/S0950-7051\(01\)00090-9](https://doi.org/10.1016/S0950-7051(01)00090-9).
21. Yurtseven, M.A., Boulougouris, E., Turan, O.: Container ship stowage plan using steepest ascent hill climbing, genetic, and simulated annealing algorithms. In: Kujala, P. and Lu, L. (eds.) *Marine Design XIII*. pp. 617–623 CRC Press (2018).
22. Zeng, M., Low, M.Y.H., Hsu, W.J., Huang, S.Y., Liu, F., Win, C.A.: Automated stowage planning for large containerships with improved safety and stability. In: *Proceedings of the 2010 Winter Simulation Conference*. pp. 1976–1989 (2010). <https://doi.org/10.1109/WSC.2010.5678873>.
23. Zhu, H., Ji, M., Guo, W.: Integer Linear Programming Models for the Containership Stowage Problem. *Mathematical Problems in Engineering*. 2020, 4382745 (2020). <https://doi.org/10.1155/2020/4382745>.
24. Review of Maritime Transport 2020, [https://unctad.org/system/files/official-document/rmt2020\\_en.pdf](https://unctad.org/system/files/official-document/rmt2020_en.pdf).