

Abstract

The container ship stowage planning problem (CSSP) has been studied all the time and many approaches have been applied to find out optimal or near-optimal solutions to this problem. In addition, many variant problems have been proposed and researched from different optimization aspects since there are many constraints in the course of actual shipping. The stowage stack minimization problem (SSMP) investigates a stowage planning problem when carriers have the obligation to ship all the given containers in different ports, with the objective to utilize the fewest number of stacks on the ship. The related research about SSMP can be seen in [Avriel et al., 2000], [Jensen, 2010], and [Wang et al., 2014], and they focused on the condition that there is no rehandle or shift, i.e., the problem they studied is the stowage stack minimization problem with zero rehandle (SSMP-ZR).

In this paper, we first propose the stowage stack minimization problem with K rehandles (SSMP-KR) based on the stowage stack minimization problem with zero rehandle and give the detail description about it. Like most of the literature, we highlight the constraints we study while put less emphasis on other constraints.

Next, heuristic algorithms are put forward to construct solutions to the SSMP-KR. It is worth mentioning that, our algorithms are suitable for the circular route. We use Java compilation software Eclipse to execute the code for our algorithms. For the loading strategies, two different processing methods are used to cope with two different situations whether the predetermined K meets or not, respectively. When the allowed rehandles run out, the problem becomes simpler because there is no rehandling operations that we need to take into consideration. In other words, we are just dealing with the SSMP-ZR problem under this circumstance. When there are rehandling operations allowed, we have to make decisions about how to place each single container loaded in each port to make the full use of each stack and minimize the total stacks used during the shipping. On this occasion, we need to consider the following factors: what the current port is, which port is the loading container transported to, what are the current heights of all stacks, etc. Since there are many factors to be taken into consideration, we have to go on the processing of parameters tuning and we eventually get the optimal parameters combination for this difficult situation through multiple parameter adjustment procedures. As for the unloading strategies, there are still two conditions and it depends on whether the unloading container is a blocking one or not. If it is not a blocking container, we just discharge it; If it is a blocking one, we have to unload it, reset the origin port of the unloading container and load it again. Our output for each instance using this algorithm in Eclipse includes the number of used stacks, the lower and upper bound number of used stacks, the number of used rehandles

and the allowed rehandles, and the layout of ship after one loading or unloading operation.

Then, we have discussed and proved the theoretical performance guarantee of the algorithm using the math knowledge of inequation and mathematical induction. We find out the lower bound and upper bound to make sure the solutions we get are reasonable and effective, which proves the feasibility of the algorithm theoretically. In actual, we use a tip in the process of proof, we divide the stacks used into two parts: the full stacks and the partial stacks. By doing so, we can deduce the lower and upper bound faster and simpler. To evaluate the actual performance of our algorithm, we conduct experiments on a set of instances with practical size and compare the results with different values of K , and of course zero is included. We have selected a set of values of K to find out the relationship between K and the stacks we used. Through this process, we may get the approximate value of K for one particular shipping to balance the flexibility and space utilization of the ship. Whats more, instances with different number of ports, containers and heights all have been experimented to make sure our algorithms have a universality.

The results demonstrate that our heuristic approaches can generate promising solutions if we allow some rehandling operations compared with the zero rehandle solutions and random loading solutions. That is to say, the results we get use less stacks in average and it shows the problem we put forward is of practical significance and it can improve the utilization ratio of vessels with certain flexibility. It is worth mentioning that our problem with K rehandles constraint acts as a buffer and offers great flexibility for the actual problem as a result of the existence of rehandling operations. Applying this algorithm into the real shipping management can enhance the hull space utilization and reduce the unnecessary spending in the case of meeting other constraints. What's more, the novelty and practicality show the importance of this paper and it can give a reference to the future study.

References

- [Avriel et al., 2000] Avriel, M., Penn, M., & Shpirer, N. (2000). Container ship stowage problem: complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics*, 103(1), 271–279.
- [Jensen, 2010] Jensen, R. M. (2010). *On the complexity of container stowage planning*. Technical report, Tech. rep.
- [Wang et al., 2014] Wang, N., Zhang, Z., & Lim, A. (2014). The s-towage stack minimization problem with zero rehandle constraint.

In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 456–465).: Springer.