Optimization Models for the Containership Stowage Problem

Peer Giemsch[1] and Andreas Jellinghaus[2]

[1] Universität Karlsruhe (TH), Fakultät für Wirtschaftswissenschaften, Institut für Anwendungen des Operations Research, Kaiserstr. 12, D-76128 Karlsruhe, Germany, peer@giemsch.de
[2] Universität Karlsruhe (TH), Fakultät für Wirtschaftswissenschaften, Kaiserstr. 12, D-76128 Karlsruhe, Germany, andreas.jellinghaus@inka.de

Abstract. This paper deals with the containership stowage problem. Containers are placed on the ship in a last-in-first-out manner and therefore temporary unloading and reloading in subsequent ports along the route, called shifting, is common and results in high costs. This is true, in particular, if the stowage plan is based only on stability constraints of the ship.
The generating of such schedules depends on the transportation load, the technical constraints and possibilities in the ports, the ship geometry, the sequence of ports visited and some other rather technical constraints (e. g. very heavy containers or hazardous goods).
We will show how this containership stowage problem can be modeled as a mixed integer programming model and discuss the computational complexity of the problem. Based on these results, solution methods are developed and some special cases are analyzed.
Furthermore, as a cross reference, we draw our attention to other combinatorial problems like the three-dimensional packing problem with special precedence constraints or pile-up problems. Finally, we propose possible directions for further research.

1 The Containership Stowage Problem

Containerships call at many ports and at each port containers are loaded and unloaded. The containers on board of a containership are put into stacks. A container is only accessible if it is on the top of the stack (last-in-first-out). The task of determining a good container arrangement is called stowage planning, e.g. [1]. Because modern containerships carry up to 7000 containers and visit up to 20 ports, optimization requires solution of a large scale combinatorial problem. To come up with a good stowage plan there might be multiple objectives that could be pursued. We attempt to minimize the container shifts, but there are other objectives as well like (see [14]):

- minimizing usage of ballast water,
- minimizing torsional and shear forces,
- maximizing utilization of the terminal equipment,

- minimizing trim,
- effective hatch usage.

In addition to the accessibility constraint named above, a huge set of other constraints, such as manitain ship stability, requirements for the storage of hazardous cargo, deck strength limits, electric supply of refrigerated units, and limited mixture of 20' and 40' containers.
Also, there are some more limitations for the container terminal and in general the loading of a ship begins before all containers to be shipped have arrived. This makes the problem stochastic and results in several updates of the stowage plan.

2 Basic Model

In the basic model (as given e. g. in [4]) the containership is modeled as a single bay as an array with R rows (heigth of stacks) and C columns (total number of stacks). There are N ports. The ship starts with an empty bay at port 1 and at port N all containers will be unloaded. The placement of the containers remains unchanged between port i and port $i+1$. We call the bay uncapacitated if $R = \infty$, capacitated otherwise.
The transportation load is given as a $(N-1) \times (N-1)$ matrix $T = (T_{ij})$, where T_{ij} is the number of containers orginating at port i with destination j. There is one standard size of container, which can be placed on arbitrary positions in the shipbay. The transportation load should be feasible in respect of hold and deterministic at every port.
There is one crane at every port which can lift any container on the quay and on the bay without limitations. All containers orginating in that ports, respectively, have arrived before the ship. The shifting cost per container is equal to 1.

Definition 1 ([1]). A container u of a column is *overstowed* when it blocks the retrieval of another container v and the destination port j of u is later on the schedule than the destination port i of v.

If there are overstowed containers *shiftings* become necessary, i. e. to temporarily unload the containers and reload them later.

Definition 2 (uCSP). Given a transportation matrix, a bay, and a nonnegative integer s, the *unrestricted containership stowage problem (uCSP)* is the decision problem if there exists a stowage plan that cause at most s shifts. Associated optimization problem is to minimize shifts.

The term "unrestricted" means, that there are no technical restrictions to be considered in a sense that each container can be placed in any empty slot.

Theorem 1 ([4]). *Let C be the number of columns in an uncapacitated bay. Then, the uncapacitated, unrestricted containership stowage problem is $\mathcal{NP}$-complete for $C \geq 4$.*

There are some special cases. Clearly every bay with $N-1$ uncapacitated columns can result in a zero-shifts plan for every transportation matrix, but it can further be reduced:

Theorem 2 ([4]). *Let T be a $(N-1)\times(N-1)$ transportation matrix with no zero entry on or above the diagonal. Then the minimum number of uncapacitated columns C^* needed for a zero-shifts plan to exist is equal to $\lceil \frac{N}{2} \rceil$.*

The common formulation of the uCSP is (see in [2]):

$$Min \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{v=i+1}^{j-1} x_{ijv}(r,c)$$

subject to

Shipquant $(i=1,\ldots,N-1, j=i+1,\ldots,N)$:

$$\sum_{r=1}^{R} \sum_{c=1}^{C} \sum_{v=i+1}^{j} x_{ijv}(r,c) - \sum_{k=1}^{i-1} \sum_{r=1}^{R} \sum_{c=1}^{C} x_{kji}(r,c) = T_{ij} \tag{1}$$

Slot $(i=1,\ldots,N-1, r=1,\ldots,R, c=1,\ldots,C)$:

$$\sum_{k=1}^{i} \sum_{j=i+1}^{N} \sum_{v=i+1}^{j} x_{kjv}(r,c) = y_i(r,c) \tag{2}$$

OnTop $(i=1,\ldots,N-1, r=1,\ldots,R-1, c=1,\ldots,C)$:

$$y_i(r,c) - y_i(r+1,c) \geq 0 \tag{3}$$

Shift $(j=2,\ldots,N, r=1,\ldots R-1, c=1,\ldots,C)$:

$$\sum_{i=1}^{j-1} \sum_{p=j}^{N} x_{ipj}(r,c) + \sum_{i=1}^{j-1} \sum_{p=j+1}^{N} \sum_{v=j+1}^{p} x_{ipv}(r+1,c) \leq 1 \tag{4}$$

$$x_{ijv}(r,c) \in \{0,1\} \qquad y_i(r,c) \in \{0,1\}$$

We define $x_{ijv}(r,c) = 1$ if there is a container in slot (r,c) loaded on board in port i with final destination j, and unloaded in port v and 0 otherwise. Furthermore $y_i(r,c) = 1$ if upon sailing from port i, slot (r,c) is occupied by a container and 0 otherwise.

The constraints allow for the following interpretation: (1) specifiy the number of containers to be shipped. (2) ensure that there is at most one container in each route segment. (3) are needed to ensure that the containers are stored in stacks and (4) define shifting movements. The objective counts the shiftings caused by overstowed containers.

Other formulations of the CSP as a mixed integer program can be found in [5] and in [11], but there are even more variables and constraints in use.

3 Extensions of the Basic Model

There are a lot of potenial expansions of the basic model, in particular with regard to technical constraints. For example, heavy containers should be loaded in the columns first. Or the containers which need electricity have to be stowed separately.
Different costs of container movements in ports can be incorporated introducing weigths into the objective function.
The provision for the ship geometry can be carried out through the usage of virtual containers to be unloaded in a port $N + 1$ or an appropriate data structure other than a matrix for the bay plan.
Conspicuous it is that stowage plans generated on the basic model tend to group containers with the same destination. This is certainly a plausible way if there is only one crane at a terminal. But it might be unfavorable in cases where more than two cranes are available because the grouped containers are only accessible by one crane.
In [12] an integrated approach which combines stowage planning and selection of loading sequences of the quay cranes is presented. Especially the just-in-time arrival of containers at the terminal is modelled.
In all cases mentioned above the basic model is not suited without arranging columns in two dimensions.
We augment the approach in that we include everal additional constraints and we examine how given solution methods and data structures have to be modified.
First: Ships are never completely empty. This makes it usually impossible to arrange all containers in order leaving the first port.
Second: Ports visited are not all different as some ports are visited twice, e. g. a typical tour between Europe and Asia is calling Shanghai and Hong Kong two times.
Third: Ships make round trips. Therefore the transportation matrix cannot be defined as above mentioned as there will be no transports between port i and $i + N$ or other pairs.

4 Solution Methods

Due to the fact CSP is $\mathcal{NP}$-complete and the formulations as a binary linear program does not provide us with much hope to obtain results in reasonable time developing heuristics seems suggestive. There are some methods for producing solutions like simulation, heuristics, rule-based expert systems and decision support systems. We want to mention here [15] as a good starting point. Wilson et al. in [15] and [14] use a two stage process of computerized planning i. e. generalized placement strategy as well as specialized placement procedures. They incorporate many of the technical constraints in the objective function and their procedure models the human planner's conceptional

approach.
We examined the basic model with our own extensions mentioned above.
The generation of random instances was conducted in the following way: Departing from the empty transportation matrix, a given number of ports, and the capacity of the ship generation of the transportation matrix is done by adding several transports. Input factors are the number of transports to add, and the maximum size for an additional transportation.
A single transportation is added by selecting a random valid (i, j) combination $(j > i)$ and adding containers. The number of containers is set to a random value from one to maximum for an additional transportation. If that amount of containers can be added to the transportation matrix without exceeding the ship capacity, it is done. If not, a new random value is selected up to the amount that can be added without exceeding the ship capacity, then that new amount of containers is added to the transportation matrix.
An alternative generation method tries to approximate real world transportation problem. Suppose that several ports are given, and the route of the containership is given as a string of indices to these ports, let n be the length of the route. The container ship is expected to cruise that route several times, the generation process approximates this loop on the given route m times. The whole journey covers $N = n * m$ ports.
Further, for example, a ship on a route linking Europe and China is not expected to carry many containers for transportation within Europe only or within China only, and is not expected to transport containers from Europe to Europe via China. Such restrictions are implemented with a port to port matrix containing weighting factors for each combination of source and destination port for the given route.
As with the simple generation of the transportation matrix, a number of transports are added to this matrix. A new transportation is selected with a random starting port $i \in \{1, \ldots, N-1\}$ and a random destination port $j \in \{i+1, \ldots, i+n-1\}$. The port associated with the indices i and j are looked up. If the vessel will reach the destination port after i and before j, then no transportation is added. Otherwise a transportation is added in the same manner as in the above named simple generation, but using the additional weighting factor from the port to port matrix. In case the weighting factor is 0, no transportation is added to the transportation matrix.
The generated instances can be downloaded from the web [8].
The stowage plans for all transports steps are build up in parallel by the following three steps of our heuristic. The first step assigning whole columns (inspired by [2]). All steps start processing transportations with maximum distance first. All transportation amount in the transportation matrix T are divided by the row height R. The numbers of whole columns a transportation can fill are noted for processing in the first step, the remaining transportation amount is left in the transportation matrix for further processing in the second and possibly third step.

At the first step whole columns are filled with containers assigned to the same cell for the whole journey. As columns are filled with containers of the same source and destination port only, no shifting is necessary. This step always succeeds, but can leave some columns fragmented in transportation steps, as the column is filled with containers only for some transportation steps but not for all.
At the second step the heuristic tries to fill the remaining containers into columns without causing shifts. Containers are assigned to partially filled columns containing containers with the same destination and the same or an earlier source port. Thus no shifts are created assigning the containers on top of the already assigned containers. Otherwise containers are assigned to columns empty during the required time, so the container can be assigned without the need for shifts.
Remaining containers are filled in the third step one by one: for every container the best assignment is calculated with a branch-and-bound algorithm to minimize shifts necessary to store that container. This algorithm selects several time slices, where there is at least one column for every time slice with a free cell during the given slice, and the time slices add up to the transportation duration of the container to be assigned. Combinations with smaller intersections of the time slices are prefered.
Containers to be shipped from port i to port $i + 1$ can never cause a shift, if these containers are placed last on top of all other containers and are removed first. The first and second step do not assign such transportations.
We compared its results with the suspensory heuristic which was implemented according to the outline Avriel et al. [3] . We could impove results in 1038 out of 1050 problem instances. Our results are available online at [8].
As placement of a column in ship bay is not specified, it is further possible to try to meet stability constraints via adjustment.

5 Related Problems

In this section we show some other problems connected with the CSP. A discussion of Tower of Hanoi problem(see [9] and for an application [10]) or sorting permutations by stacks (see [7]) are omitted due to the scope of this paper.

5.1 Tram Dispatching

In a recent PhD-Thesis [16] the tram dispatching problem is analyzed. In this problem it has to be decided how incoming trains have to be distributed to tracks in the depot so that no shunting occurs in the next morning. The difference to the CSP is that no trains leave the depot before all trains have arrived and this difference is crucial. The thesis additionally analyses the online case. The online case means just in time changes as crash of trains and delays and are like result changes in the stowage plan if new containers arrive to load. So this approach can possibly be adapted to the stochastic CSP.

5.2 Packing with discharge conditions

The vehicle routing problem with packing constraints was recently presented [13]. In this problem, there are some vehicles with dynamic pickup and delivery at the customers and the gemoteric load dimensions are also considered and not only the total weight or volume of boxes to be picked. The goal is to minimize tour length and unutilized volume.

In a new approach we analyze the loading and unloading operations by each delivery. Because the volume of each vehicle is limited and there is normally only one access point at the rear, it may be necessary to unload some boxes temporarily. These shifts depend on the sequence of the visited customer and the packaging pattern.

In general we consider a three-dimensional packing problem according to Dyckhoff's [6] classification as $3/V/O/C$. This denotes a three-dimensional packing problem where one large object has to be packed with all items of congruent figures. If every item has a destination and there is a certain sequence of destinations it is useful to think about the discharging procedures. One can identify two cases:

In the first case all items have to be loaded in the beginning. Then, the objective is to minimize, for example, the unutilized space and to minimize shifting operations at each destination by unloading the items.

The second case is more complicated when loading and unloading occurs at every destination as is the case in vehicle routing problems with packing constraints explained above.

In contrast to CSP, the items are not loaded in stacks and there might be very complicated patterns of packed items. The boxes are overstowed in different manners, e. g. there are overstowed boxes above but also in front or beside a box. Clearly there is only one type of boxes in CSP.

6 Further Research

The complexity of the uCSP is still unknown for fixed number of rows R. Also it can be shown that for a single uncapacitated column there exists a polynomial time algorithm [1], the complexity of the uCSP remains unsolved for $C = 2$ or $C = 3$.

There is no application of these models and solutions methods to real data. Many solution methods seems to be ad hoc so that a comprising analysis is auspicious.

A new formulation as a mixed-integer program which can be expanded to multicriteria objectives would be eligible.

References

1. Aslidis, A. (1990) Minimizing of Overstowage in Container Ship Operations. Operations Research **90**, 457-471
2. Avriel, M. and Penn, M. (1993) Exact and Approximate Solutions of the Container Ship Stowage Problem. Computers and Industrial Engineering **25**, 271-274
3. Avriel, M.; Penn, M.; Shpirer, N. and Witteboon, S. (1998) Stowage Planning for Container Ships to Reduce the Number of Shifts. Annals of Operations Research **76**, 55-71
4. Avriel, M., Penn, M. and Shpirer, N. (2000) Container Ship Stowage Problem Complexity and Connection to the Coloring of Circle Graphs. Discrete Applied Mathematics **103**, 271-279
5. Botter, R.C. and Brinati, M.A. (1992) Stowage Container Planning: A Model for Getting an Optimal Solution. In: Proceedings of the IFIP TC5/WG 5.6 Seventh International Conference on Computer Applications in the Automation of Shipyard Operation and Ship Design, VII Rio de Janeiro, Brazil, 10-13 September, 1991, 217-229
6. Dyckhoff, H. (1990) A Typology of Cutting and Packing Problems. Eurepean Journal of Operational Research **44**, 145-159
7. Even, S. and Itai, A. (1971) Queues, Stacks and Graphs. In: Kohavi, Z. et al. (1971) Theory of machines and computations. Academic Press, New York London, 71-86
8. Giemsch, P. and Jellinghaus, A.: Test-Instances for the CSP. http://www.andor.uni-karlsruhe.de/fak/inst/andor/a10_www/csp.html (last visited 2003-09-05)
9. Hinz, A. M. (1989) The Tower of Hanoi. Enseignement Mathematique, II. Séries **35**, 289-321
10. Sarkar, U.K. (1998) Solution to Cargo Loading Problem Using Multipeg Towers of Hanoi - A Heuristic Search Approach. In: Proceedings of the International Conference on Knowledge Based Computer Systems (KBCS-98), Mumbai, India, 65-76
11. Schott, R. (1989) Stauplanung für Containerschiffe. Vandenhoeck u. Ruprecht, Göttingen (in german)
12. Steenken, D. and Winter, T. and Zimmermann, U.T. (2002) Stowage and Transport Optimization in Ship Planning. In: Grötschel et al. (2002) Online Optimization of Large Scale Systems. Springer, Berlin Heidelberg, 731-745
13. Türkay, A. and Emel, E. (2003) Vehicle Routing Problem with Packing Constraints. Presentation on the 5th Euro/Informs Joint International Meeting, July 06-10, Istanbul
14. Wilson, I.D. (1997) The Application of Artificial Intelligence Techniques to the Deep-Sea Container-Ship Cargo Stowage Problem. PhD-Thesis, Conta University of Glamorgan - School of Accounting and Mathematics, Glamorgan
15. Wilson, I.D. and Roach, P.A. (2000) Container Stowage Planning: A Methology for Generating Computerised Solutions. Journal of the Operational Research Society, **51**, 1248-1255
16. Winter, T. (1999) Online and Real-Time Dispatching Problems. PhD-Thesis, Technische Universität Braunschweig - Fachbereich Mathematik und Informatik, Braunschweig