Development of reuse module on stowage planning system using CBR for multiport route

N L A Pramesti1, I Mukhlash2, and S Nugroho3
1,2Department of Mathematics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
3Department of Marine Transportation Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
E-mail: 1lailiastip@gmail.com, 2imamm@matematika.its.ac.id, 3s.nugroho@seatrans.its.ac.id

Abstract. Stowage planning is an activity of containers arrangement on container vessels. A good and fast container setup requires a piece of software that can help planner’s performance especially for multiport cases. The Case Based Reasoning method (CBR) can be used to implement software in order to produce a container setup based on pre-existing setups. The Case Based Reasoning method discussed in this paper is focused on the reuse phase by designing the Similarity algorithm, Container Loading List adaptation algorithms as well as the restow evaluation algorithm. The set of algorithms is used to automatically produce a container setup at each port in one voyage and display the number of restow and many containers that experience loading and unloading in one voyage so as to assist the performance of the planner in every port. The results of this study can also be analysed for the workload of planners where the smaller the case base population, manual swaps performed on average more than 70% of the total containers on the ship.

1. Introduction
Indonesia is an island nation where $\frac{2}{3}$ of Indonesia is an ocean. Also supported by the development of industrial processing today in various regions to improve the regional economic strategy, making the potential demand for shipping services by using sea transportation in the form of containers is much in demand by industrial entrepreneurs. Currently, more than 70% of industrial companies choose to ship goods using large amounts of sea transportation. This makes the port very important in influencing the quantity and size of ships that can lean on the docks. The port must handle larger ships and carry a greater number of containers in a short period of time. In many cases, stowage planning is often fickle even two hours before the loading process is complete. This is due to changes to the container loading and unloading list (CLL), ranging from the addition of loaded containers to the cancellation of container unloading. This situation leaves stowage planning vulnerable to human error. The human error caused greater port operational costs and the time it took for the ship to operate also longer due to the restow process which is the process of restructuring the container position on board for the efficiency of the ship’s space. For multiport cases, the arrangement of containers to the ship does not only take into account the size and weight of the container. But also the location of the port where the container was sent. Container arrangements should take into account the route of the ship’s journey to minimize the movement of cranes during the loading and unloading process at each port as the movement
of cranes can affect the total cost that must be borne at each port. To solve this problem, it takes software that can help planners so that stowage plans can be quickly created for all ports even if the CLL is fickle. The design of the stowage plan can be created by equating the existing stowage plan that bears the resemblance of CLL. For that it is used the Case-Based Reasoning method in resolving this case. Case-Based Reasoning (CBR) is a method of troubleshooting approaches using past problem solving experiences that bear the highest resemblance. So that one can analyse one or more solutions from the previous relevant problem to find a new solution to the problem.

The Case-Based Reasoning method has four stages. The first stage is Retrieve where at this stage it begins sorting and selecting past problems that have had a resolution and bear the highest resemblance to the problems resolved today. This past and solving problem will be used for the second stage, the Reuse stage. The Reuse stage is to adapt the current container unloading list to resemble the problem selected in the previous stage. Then, the third stage is the Revise stage where a review of the adaptation results has been done in the Reuse stage in order to produce a solution to the new problem. The Revise stage does not have to be done, following the conditions of the Reuse stage results. Furthermore, the last step is Retain which is useful for storing the latest issues and their workarounds into the Database so that they can be used to solve subsequent problems.

In previous research, prototype stowage planning system modules have been created that can integrate with iStow software. This research was conducted by Nurdin Akbar Herwanto in 2011, “Development of a Case-Based Stowage Planning System Module Prototype for Semi-automatic Container Stowage on Ships” which is the retrieval stage implementation of the Case-Based Reasoning method. Further research was conducted by Agung Pratama in 2019, “Design and Application of Reuse Stage Algorithm of Case-Based Reasoning Method on Container Stowage” which produced software that could facilitate the arrangement of containers for loading by using the Case-Based Reasoning Method at the Reuse stage. However, in the study, the software can only be used for loading process by transporting containers to one port only. Thus, the authors are interested in developing the research to facilitate the system of structuring containers on multiport routes. The Case-Based Reasoning method is used in the implementation of a set of Reuse stage algorithms designed on the software to obtain the number of restow, manual swaps and program swaps on the case base population to analyse the impact of the planner’s performance load in performing container setup on the software.

2. Literature review

2.1. Case-based reasoning

The Case-Based Reasoning method is one of the methodologies used to find problem solving by leveraging experience for previous problems that are almost identical to the issues being discussed. Generally, this method presented a problem, this problem could come from a system or program. Then, using this method, start looking for past issues that have been resolved and have the same problem specifications as the new problem to be resolved. If there is no similar problem, there will be some problems similar to the new one [5]. The Case-Based Reasoning method has four stages: Retrieve, Reuse, Revise and Retain. In the Case-Based Reasoning Method, the more data stored in the database, the resulting solution to the new problem is better than if the data stored in the database is less, then the results or solutions obtained to solve the new problem are also not better. The flowchart diagram of the Case-Based Reasoning Method can be shown in Figure 1.
2.2. Casestow

Casestow is an implementation of the Case-Based Reasoning Method used in stowage planning, as the stowage planning process becomes semi-automatic. This method can of course ease the burden of planners. When the stowage planner gets a new problem, then he will start utilizing the experience to solve the previous stowage planning problem to find a solution to the new stowage planning problem. The point is to be able to manage the experience to do the planning, so that the process of retrieving and reuse a case can be executed. Stowage plan issues can be described as follow

Input of the process:

< stowage plan prior to loading, container loading list >

Solution of the process:

< departure stowage plan >

Every planning session may produce stowage plans of a variety of quality, from a poor to an excellent one. This information is stored as a comment on the plan

Remark:

< comments on departure stowage plan >

Case:

< stored problem, stored solution, stored remark >

Query:

< actual problem >

Case is a finished planning session, it will be example to make new stowage plan for query. Query is a new stowage planning task or an actual problem [6]. The flowchart diagram of the casestow process can be shown in Figure 2.
2.3. Row-bay-tier coordinate concept

The row-bay-tier coordinate concept is used to facilitate the visualization of the container setup process. The row-bay-tier concept follows a numerical coordinate system related to length, width and height. Bay is a numerical coordinate system of container setup for transverse direction. Row is a numerical coordinate system of container setup for elongated rows. Tier is a numerical coordinate system of container setup for vertical layers or stacks.

2.4. Multiport concept

Multiport cruise routes occur where ships visit more than two ports on their voyages. During the cruise route, the ship stops at each port to transport and lower the container. Thus, stowage planning made for each port combines stowage planning made at the previous port with container information to be dismantled and loaded at that port, so that stowage planning at each port is interconnected. In container setup with multiport shipping lanes, the list of containers to be
loaded not only uses the parameters of the weight, size, and type of the container, but must also contain the parameters of the port of origin and port of origin of the container shipping destination.

![Figure 4](image.png)

**Figure 4.** Illustration of multiport routes through 6 ports.

### 2.5. Restow concept

Restow is the process by which containers are dropped from ships and loaded back onto the same ship in a single port or terminal location. This can happen one of them because there are containers located inside the lower vessel that have to be moved at the port. However, because it is blocked by the container above it, the upper container drop is done first, then the container underneath can be retrieved and transferred to the port. If the destination of the container port located above is different, then the container must be put back on the ship to be shipped to the destination port.

![Figure 5](image.png)

**Figure 5.** Illustration of restow concept.

### 3. Result and discussion

#### 3.1. Software specification and modelling

**3.1.1. Functional requirements**

Outlined about the things that want to be realized in the program, including:

- The development of this software is specific to KM Kendhaga Nusantara 1.
- The system can perform container setup according to the container loading list in each port automatically adapted from the selected case base.
- The system can display visualizations of the results of the order and provide information on the location of container on the ship.
- The system can calculate the number of manual swap, program swap and restow movements at each port.
• The system may free the user to change the location of the container on the result of automatic container setup when there is a container that is located in accordance with the subjective requirements of the stowage planner.

3.1.2. Use case diagram
It is used to illustrate interactions that occur between users and created programs

![Use Case Diagram](image)

**Figure 6.** Use Case Diagram.

3.1.3. Activity diagram
It is used to know workflow on the system.

![Activity Diagram](image)

**Figure 7.** Activity Diagram.
3.1.4. Flowchart of reuse step algorithm
This diagram interpreted function of container setup algorithm in each port.

![Figure 8. Flowchart of reuse step algorithm.](image)

3.1.5. Similarity algorithm
This algorithm is used to perform container setup on the new Container Loading List which has the same weight and port of destination as the container on the case base.

Input: Unorganized Container Loading List, case base retrieve stage results for each port and ship condition when arriving at port.

Step 1. For the entire container location on the case base, perform steps 2-5.
Step 2. Check the weight and port of destination of all containers on the case base.
Step 3. For the entire container location on the Container Loading List, perform steps 4-5
Step 4. Look for containers on the Container Loading List that have the same weight and port as the case base.
Step 5. Place the container in Step 4 into the process system according to the container’s location in Step 2 as long as the container location on the ship is still empty.
Output: Container Loading List container settings that correspond to the container order on the case base based on the type and port of destination of the container.

3.1.6. Container loading list adapted algorithm
This algorithm serves to fine-tune the results of similarity algorithms. Divided into two, namely the fill-blank algorithm and the algorithm add the rest of the container. Fill-blank algorithm used to repair floating containers.

Input: container loading list container settings that correspond to the container settings on the case base based on the type and port of destination of the container.

Step 1. For the entire container location with the process system, perform steps 2-3.
Step 2. Look for the location of the container that floats because there is an empty container location underneath. The search was carried out from the top of the container site.
Step 3. Look for the location of the empty container to occupy the floating container.
Step 4. For 20ft container, take step 5 and for 40ft container, take steps 6-7.
Step 5. Move the container that floated in Step 2 to the location of the empty container below it as in Step 3.
Step 6. Since the 40ft container occupies 2 locations, it is checked that both container locations in Step 2 float both.
Step 7. Move both container locations in Step 6 to two empty container locations in Step 3.

Output: No 20ft container arrangement hovers, but there is still a 40ft container arrangement floating in one of its locations when one of the two empty container locations under the floating container has been filled with other containers or one of them is not empty.

Adding Remaining Containers Algorithm is used to add containers to the Container Loading List that have not yet entered into the container order.

Input: The container order of the Container Loading List which outputs from the fill-blank algorithm and the container on the container loading list that has not been organized.

Step 1. For all containers measuring 20ft on the Container Loading List, perform Step 2, while for containers measuring 40ft on the Container Loading List, perform Steps 3-4.
Step 2. Insert an unorganized 20ft container into the process system according to the order of the groups, so that containers with heavier loads will be arranged first.
Step 3. Find two empty locations on the container order according to the container setup rules.
Step 4. Place a 40ft container into each of the two empty locations in Step 3 according to the group order, so that containers with heavier loads will be arranged first.

Output: the container on the Container Loading List has been all organized and no container stack is floating.
3.1.7. Restow evaluation algorithm

This algorithm is used to obtain the number of containers affected by restow in each port.

Input: The result of the container order from the previous port that has not been done unloading containers at the destination port

Step 1. For all containers resulting from the previous process, perform step 2.

Step 2. Looking for containers with the port of destination is the port that is being processed.

Step 3. Remove the container that has the same destination port as the port that is being processed.

Step 4. Look for containers in an order that has an empty location under the available containers. For containers measuring 20ft, take steps 5-6 and for containers measuring 40ft, perform steps 7-8.

Step 5. If there is a 20ft container that is empty underneath, then the container is hit by a restow and the value of the restow increases by one.

Step 6. For containers affected by restow will be inserted into container loading list at the port being processed.

Step 7. If there is a 40ft container that occupies two locations in the container order that has an empty location in one or both locations below, then the container is hit by a restow and the restow value increases by one.

Step 8. For containers affected by restow will be inserted into container loading list at the port being processed.

Output: A container order that has several empty locations according to the container that is unloading the container at the destination port and the addition of containers to the Container Loading List at the port being processed.

3.1.8. Satisfy algorithm

This algorithm serves to move the location of the container resulting from an automatic order if there are multiple container locations in the order that do not comply with the ship's safety requirements or do not meet the "subjective" requirements of the stowage planner.

Input: The result of a container order that does not meet the ship's safety requirements or the "subjective" requirements of the stowage planner.

Step 1. If the results of the container order do not meet the ship safety requirements, perform Steps 2-8, but if the container order does not meet the "subjective" requirements of the stowage planner, take steps 9-10.

Step 2. If the container order result is unstable or the GM value is less than zero, take steps 3-5, if the result of the order is stable but not trim eligible, or the trim value is more than 1% Length Between Perpendicular, perform steps 6-8.

Step 3. For each bay, check the weight of the container on each row and tier.

Step 4. Make a comparison between the container in Step 3 and the container located on top of the fairy pack Step 3.

Step 5. If the Step 3 container is lighter than the container located above the Step 3 container, a location conversion will be made between the 2 containers.
Step 6. Check the trim value, if the trim value is negative which means the ship is undergoing stern trim then do Step 7, but if the trim value is positive which means the ship is undergoing bow trim, do Step 8.

Step 7. Look for a heavy container on the stern of the ship and exchange it for a lighter container on the bow of the ship.

Step 8. Look for a heavy container on the bow of the ship and exchange it for a lighter container on the stern of the ship.

Step 9. Looking for container locations that do not comply with the "subjective" requirements of the stowage planner.

Step 10. Change the location of the container in Step 9 to the location that corresponds to the "subjective" condition of the stowage planner. This step could lead to an exchange between two containers or it could also be a container exchange in Step 9 with an empty location.

Output: The result of a container order that meets the ship’s safety requirements and the "subjective" requirements of the stowage planner.

3.2. Testing and analysis

The program was conducted for container setup program with multiport shipping route through four ports, namely Surabaya Port (P1), Makasar Port (P2), Belawan Port (P3), Jakarta Port (P4). The container group is divided by the size and weight of the container. There are six groups used in the program, namely group 1 for containers measuring 20ft and weighing more than 20 tonnes, group 2 for containers measuring 20ft and weighing between 10-20 tonnes, group 3 for containers measuring 20ft with a weight of less than 10 tonnes, group 4 for containers measuring 40ft weighing more than 20 tonnes, group 5 for containers measuring 40ft and weighing between 10-20 tonnes and group 6 for containers measuring 40ft weighing less than 10 tons.

3.2.1. Data testing

There are two different test data used in this study to obtain an analysis of the planner’s load and the number of restow.

1. The first test data is used to determine the effect of the planner’s performance load seen from the percentage of manual swap amounts in each port with the case base population. Manual swaps are defined as container setups that need to be done manually because there is no same type/type of container on the case. Manual swaps are calculated from the number of containers left in the CLL after container setup that has the same type as the case.

| No | No | Voyage | Query (Container Loading List) |
|----|----|--------|--------------------------------|
|    |    |        | Sby-Mks | Mks-Blw | Blw-Jkt | Jkt-Sby |
| 1  | 1  |        | 82      | 92      | 86      | 84      |
| 2  | 2  |        | 74      | 77      | 84      | 72      |
| 3  | 3  |        | 94      | 86      | 92      | 88      |
| 4  | 4  |        | 99      | 87      | 94      | 95      |
| 5  | 5  |        | 97      | 90      | 81      | 82      |

2. The second test data is used to determine the effect of the number of restow and the number of swap programs on the arrangement of containers for new cases with the amount of container contents on the case used. There is one case that has been modified the amount of contents of the container on the ship, ranging from the largest number to the least. Five different case modifications in Table 2 are used to create container setups on 4 queries that each case and query are only tested in one voyage so they are not connected.
Table 2. The number of container on board which have modified in one case.

| No | No Voyage | Case (Container Loading List) |
|----|-----------|-----------------------------|
|    |           | Sby-Mks | Mks-Blw | Blw-Jkt | Jkt-Sby |
| 1  | Case1 #0  | 100     | 100     | 100     | 100     |
| 2  | Case1 #1  | 82      | 92      | 86      | 84      |
| 3  | Case1 #2  | 42      | 72      | 86      | 64      |
| 4  | Case1 #3  | 32      | 50      | 42      | 40      |
| 5  | Case1 #4  | 10      | 15      | 16      | 19      |

Table 3. The number of container on board each query.

| No | No Voyage | Query (Container Loading List) |
|----|-----------|-----------------------------|
|    |           | Sby-Mks | Mks-Blw | Blw-Jkt | Jkt-Sby |
| 1  | Query 1   | 86      | 97      | 80      | 94      |
| 2  | Query 2   | 91      | 94      | 81      | 82      |
| 3  | Query 3   | 81      | 97      | 97      | 83      |
| 4  | Query 4   | 97      | 94      | 86      | 94      |

3.2.2. Testing process

In the trial the program can be displayed an illustration of the results of the container setup in each port and the number of containers that undergo unloading, loading, manual swaps, program swaps and the number of restow that occur in each port. Here are some examples of setup results views in each port and final views that contain setup across all ports in a single voyage.

Figure 9. Stowage planning display illustration on each port

Figure 10. Output display illustration on each voyage
3.2.3. Analysis of testing process
The testing process has been conducted on both test data with different test objectives and obtained the following results:

1. First data trial results
The first trial data was used to analyse the effect of case base population numbers on manual swap processes that planners need to do in creating stowage planning for new queries or CLL. Manual swap percentages are used to see how high the planner’s performance load is in making stowage planning. The data generated in this test is presented in the graph below.

![Manually Swap Percentage vs Case Base Population](image1)

**Figure 11.** Manually swap percentage vs case base population.

In Figure 11, it can be seen that the larger or richer the case base, the fewer manual swap percentages due to the increasing population in case base, so that in the creation of new stowage planning, planners can choose more cases used as examples in the creation of new stowage planning. Meanwhile, if the case base is small then the manual swap is getting bigger because the case in the case base is limited and does not necessarily match the new query. With the large case base population, the planner’s performance is lighter, because planners can choose the case most similar to the query so as to minimize manual swaps that need to be done.

2. Second data trial results
In the second trial, the second trial data was used to pin the effect of the number of restows and the number of program swaps on the amount of container contents on the case. The data generated in this test is presented in the graph in Figure 12 and Figure 13 below.

![Number of Restow vs Number of Container on Board Each Case](image2)

**Figure 12.** Number of restow on each query vs number of container on board each case.
In Figure 12 it can be seen that Case1 #0 which has the highest number of containers on the case compared to Case1 #1 to Case1 #4 can affect the number of restow in one voyage averaging 9 containers for four queries that were tested. Meanwhile, for Case1 #4 who had the fewest number of containers among others affected the number of restow in one voyage averaging 27 containers. Thus, it can be concluded that the more containers on the case, the smaller the restow that occurs in a single voyage. On the other hand, the fewer containers in the case, the more examples given to make stowage planning on the query are fewer so that the restow that occurs in one voyage gets larger.

In Figure 13 it can be seen that Case1 #0 which has the highest number of containers on the case compared to Case1 #1 up to Case1 #4 can affect the number of program swaps in one voyage averaging 9 containers for four queries performed testing. Meanwhile, for Case1 #4 which has the fewest number of containers among others affects the number of program swaps in one voyage averaging 17 containers. Thus, it can be concluded that the more containers on the case, the smaller the swap program that occurs in one voyage. On the other hand, the fewer containers on the case, the more examples given to make stowage planning on queries less so that the swap program that occurs in one voyage gets bigger.

4. Conclusions

Based on the results of the program trials and discussions in previous section, the data concluded that:

1. Reuse stage algorithm in Case Based Reasoning Method can be implemented on software design for container setup with multiport route.
2. The usage of algorithms in the software depends mostly on the number of case base populations and the number of containers on the case, the more the number, better stowage plan is made.
3. The software makes it easy for planners to know the number of restow in each port and the total restow in one route of travel.
4. The absence of a case base may affect the number of containers affected by restow. The larger the number of containers on the case, the smaller the restow in one voyage.
5. The number of restow on the new container setup result is influenced by the number of case base population and the number of container contents on the case used. The smaller the contents of the container in the case, the more irregular the new container order is produced so that the number of restow also increases or gets larger.
6. The case base can alleviate the planner’s performance in making stowage plan. More case base population, the lighter planner’s performance. This is judging by the number of manual
swaps that planners have to make. In the test results, if the case base population does not exist or 0, then the average manual swap percentage is above 70% of the number of containers on board and this percentage decreases as the case base population increases.

5. References

[1] Ambrosino D, Paolucci M and Sciomachen A 2013 Experimental Evaluation of Mixed Integer Programming Models for the Multi-Port Master Bay Plan Problem. Flex Serf Manuf J pp 263-284

[2] Ambrosino D, Massimo P and Sciomachen A 2015 A MIP Heuristic for Multiport Stowage Planning. Transportation Research Procedia pp 725-734

[3] Ambrosino D, Massimo P and Sciomachen A 2017 Computational Evaluation of A MIP Model for Multiport Stowage Planning Problems. Soft Comput pp 1753-1763

[4] Harwanto N A 2011 Pengembangan Prototipe Modul Sistem Perencanaan Stowage Berdasarkan Kasus untuk Penataan Semi-Otomatis Peti Kemas pada Kapal. Thesis Mathematics Department, Institut Teknologi Sepuluh Nopember (Surabaya)

[5] Main J, Dillon T and Shiu S 2001 A Tutorial on Case Based Reasoning: Soft Computing in Case Based Reasoning (Eds). Sprenger-Verlag pp 1-28

[6] Nugroho S 2004 Case-based Stowage Planning for Container Ships in International Logistics Congress 2004 (Izmir, Turkey: Dokuz Eylul University Publications) pp 609-618

[7] Nugroho S 2005 CASESTOW: Recycling of Past Stowage Plans in 1st International Conference on Operations and Supply Chain Management (Bali, Indonesia)

[8] Pratama A 2019 Design and Application of Reuse Stage Algorithm of Case-Based Reasoning Method on Container Stowage Planning In Journal of Physics: Conference Series vol. 1490, (Indonesia: IOP Publishing)

[9] Riesback C and Schank R 1989 Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Inc.

[10] Wilson I and Roach P 2000 Container Stowage Planning: A Methodology for Generating Computerised Solutions in Journal of the Operational Research Society pp 1248-1255