Container Pre-marshalling Problem: a Review and Solution Framework

Mohamed Gheith
Egypt-Japan University of Science and Technology

Container Pre-marshalling Problem: A Review and Solution Framework

M. S. Gheith*1, A. B. El-Tawil*2, N. A. Harraz*3

*Department of Industrial Engineering and Systems Management,
Egypt - Japan University of Science and Technology (E-JUST), Alexandria, Egypt
1mohamed.gheith@Ejust.edu.eg, 2eltawil@Ejust.edu.eg, 3nermine.harraz@Ejust.edu.eg

Abstract - **The continuously increasing trends of global trade are putting more emphasis on container terminals related operational problems. These problems are becoming day after day of extreme importance, and ever improving performance has become a demand for all container terminals. The container pre-marshalling problem is one of the important and hard to make decisions in modern container terminals that has a direct impact on vessel berthing times and hence on the performance of the container terminal. In this paper we provide a review of recent literature and we propose an innovative simple heuristic.**

Keywords - **Container pre-marshalling problem, container retrieval problem, container stacking problem, container yard terminals, heuristics.**

I. INTRODUCTION

In container terminals three areas exist: the quay side, the land side, and the container yard. One of the most important container terminals performance indicators is the berthing time of vessels [1]. It is desired to minimize the berthing time of the vessels, which, does not only depend on the performance of the Quay Cranes (QC), but also depends on the performance of Yard Cranes (YC). The work rate of the QC is double that of the YC [2], and thus, YC operations are potential bottlenecks. This low rate results mainly because the containers are not stacked in the yard in a configuration that facilitates the work of the QC, so that, the container pre-marshalling problem is of importance in container terminals.

As shown in Fig. 1, a container yard consists of blocks, each block consists of a set of bays, each bay consists of a set of stacks, and each stack consists of a set of tiers. When containers arrive to the yard, they are piled up above each other waiting for further operations. In order to get a container that is not on the top of the bay, additional movements have to be made to change the location of the containers that are above the desired container in order to get the desired one. These additional movements are called "reshuffles" which consume too much time leading to reduced work efficiency of YC and QC especially during vessel loading operations.

Generally, the container stacking problem is classified into three main types: the pre-marshalling problem, the remarshalling problem and the container retrieval problem [3]. The pre-marshaling problem, is the problem of converting an initial layout of a bay into a desired final layout within which containers are stacked above each other with the priority of stacking the containers that will be served first at the top of the stack. This will minimize or eliminate future additional reshuffles by the YC. In the case of the container retrieval problem, it is desired to remove a container from the bay with minimum number of reshuffles, and then remove another container and so on till the bay is empty. The three classes of the problem are of prime importance knowing that in large container terminals, the average number of movements made by yard cranes is 15,000 movements per day [3], which means that the reduction of such moves will dramatically improve operations and efficiency. Following this introduction, the second section of the paper includes a review of the literature of the past few years. In the third section, a proposed heuristic for solving container the pre-marshalling problem is presented. Finally, the conclusions and directions for future research are discussed in the fourth section.

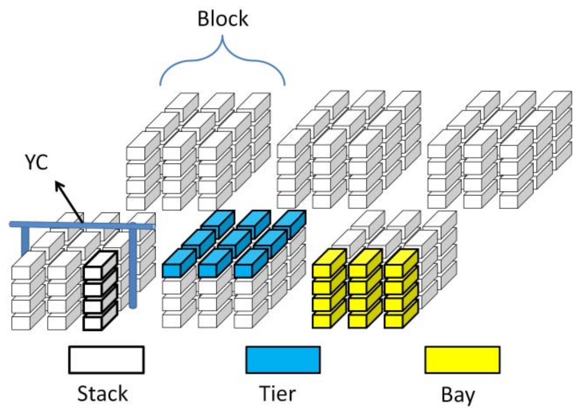

Fig. 1. Layout of storage area of a container terminal

II. REVIEW OF LITERATURE

One major problem that we faced when reviewing the container pre-marshalling problem is that, the literature discussing this problem is very limited [4]. Also, it is difficult to classify the literature according to the methodologies used, because most of papers used heuristics and approximate algorithms as the problem is NP-hard [5]. In this paper, we track the previous work of the different authors in chronological order to elaborate the progress and improvement of their work.

Stahlbock and VoB, provided an extensive review about container terminals operations in [1], which they updated later in [3]. In the section of the pre-marshalling problem, the review did not change. They described the environment of the pre-marshalling problem and the constraints that restrict it. Also, they provided a review about the work done in pre-marshalling problem but without a classification.

Bortfeldt, and Foster [6], provided a tree search heuristic for solving container pre-marshalling problem. The search was carried out by means of recursive procedure, and the procedures were coded using C programming language. The authors provide a comparison between the results obtained by their own heuristic and the results obtained by the algorithms of Lee and Hsu [7] and Lee and Chao [8] and in both cases, the proposed tree search heuristic was better.

Lee and Hsu [7], proposed an integer programming model to solve pre-marshalling problem for small and medium size instances. In large size problems, the authors developed a simple heuristic that is capable of providing a solution while minimizing computation time. But, the authors did not give any comparison between their method and any other methods.

Lee et al [9], proposed a three phase heuristic based on heuristic rules to solve the container stacking problem. First, it generates an initial feasible movement sequence according to a simple rule. The second phase quickly reduces the total number of required movements in the sequence by repeatedly solving a binary integer program. In the third phase, the heuristic uses a mixed integer program to reduce the total working time the crane needs to complete the entire sequence. Computational experiments show that, the proposed heuristic is able to solve instances with more than 700 containers.

Caserta et al [10] developed three methods of solving the container relocation problem. The first one is the development of a binary integer programming model which is capable of solving small-size problem instances. Due to the large computation time, the model was simplified by some realistic assumptions to be able to solve medium-size problem instances. For large-size problem instances the authors developed heuristic rules based on the computation of a stack score, which helps to determine where a relocating container should be placed, the heuristic was coded in C++. Also, Caserta et al [11], proposed another heuristic approach for solving the problem of relocating containers, the proposed heuristic was based on the Corridor Method (CM). The basic idea of the CM is to use an exact method (e.g. Dynamic Programming (DP)) over a restricted portion of the solution space of a given problem in order to minimize the solution space of the problem. The authors used a modified DP algorithm as an exact method with CM. They solve their problem based on the assumptions of Kim and Hong [12]. In order to reduce the search space of the problem, two parameters were defined concerning the height of the stacks and the number of the stacks. The algorithm was coded in C++ and computational experiments were made. Compared with the results of Kim and Hong, their proposed algorithm showed a decrease in the number of movements required in small to medium size instances, while in large instances, the proposed algorithm shows a decrease in the number of movements required, but, in a relatively larger time. In case of very large instances, the algorithm showed a tremendous decrease in the required number of movements, but, in very large amount of time compared with Kim and Hong.

Extending their work in [13], Molins et al [14], presented a new domain-dependent planning heuristic to solve the container stacking problem. The heuristic was coded using the Planning Domain Definition Language (PDDL) language. The advantage of this language is that, it is capable of representing the physical characteristics of the objects under study. The results of their heuristic were compared with the results of their old one [13], and the results show a significant decrease in computation time.

Also, Molins et al [15], tried to integrate the container stacking problem with the Berth Allocation Problem (BAP) and the Quay Crane Assignment Problem (QCAP). First they solve the container stacking problem with a method they developed earlier [14], after that they solve the integrated BAP+QCAP problem, then they proposed a planner that integrated the solution of the two problems, and the terminal operators are to decide which solution is the most appropriate in relation to a multi-objective function.

Exposito-Izoquierdo et al [16], illustrated the Lowest Priority First Heuristic (LPFH) used to solve the pre-marshalling problem based on the assumption of Lee and Hsu [7]. Also they introduced a generator to generate instances for the pre-marshalling problem. The generator is capable of generating different size instances (small, medium, and large). The heuristic was coded using Java programming language, and the results were compared with the results obtained by Caserta and VoB [10], and it was found to be relatively perfect. Also, the results were compared with the optimum solution gained by A^* search algorithm, and the results demonstrate the good performance of the proposed heuristic.

Sauri and Martin [17], presented different stacking strategies for only outbound (import) containers. The performance of their strategies was tested using a mathematical model and the results of the different strategies were compared. It is worthy to say that, the instances used in such research were small instances, and that is may be due to the NP-hard nature of the problem.

Park et al [18], provided an online search algorithm that is capable of optimizing stacking operations in an Automated Container Terminal (ACT). The authors discuss the difference between the stacking problem in ACT and in conventional container terminals. One of the advantages of their proposed Dynamic Policy Adjustment (DPA) algorithm is that, it is capable of investigating a candidate solution while at work. The authors used a vector weight performance measure to check the performance of the DPA algorithm using simulation experiments.

Huang and Lin [3], define two types of the container stacking problem. The first one is the same as the traditional stacking problem, which is the stacking of the containers in order to minimize the total number of reshuffles. The second problem is the grouping of containers into the yard as a set of groups, each group has similar characteristics (e.g. dangerous containers, hazard containers, ..., etc). The authors proposed two heuristics (heuristic-A, and heuristic-B) to solve the two problems. Both heuristics are based on labeling algorithms that label the stacked containers to decide next movement. In order to check the performance of the proposed heuristics, computational experiments were done. Compared with the solution obtained by Lee and Hsu [6], heuristic-A shows a decrease in the number of required movements for large problem size.

Hou et al [19], proposed a hybrid algorithm based on constraint satisfactory and best fit-approximation to solve the container stacking problem. The results from the proposed algorithm were compared with the results obtained from Best - Fit Decreasing (BFD) algorithm.

As a conclusion from the literature review, the mathematical models are capable of solving only small instances, while in large problem instances heuristics are appropriate. Also, there are limited comparisons between the results obtained from different solution methods. Based on these remarks, we developed a heuristic to solve the problem under study, and tired to avoid these remarks in the future work.

III. THE PROPOSED HEURISTIC TO SOLVE THE PRE-MARSHALLING PROBLEM

The proposed methodology is based on developing a five stages heuristic in order to minimize the required movements with short computation time. In case of small instances, integer programming model will be used and then, the results of the proposed heuristic will be compared with this model. For validation, in case of large instances, the results obtained from the proposed heuristic will be compared with benchmark problems found in the literature, and the performance indicators will be calculated. As described before, it is required to rearrange the initial layout of the containers so as to get a final desired layout that is confirmed with the stowage plan of the vessel. For safety considerations, the yard crane usually serves one bay [1]. Thus, our heuristic will focuses on solving the pre-marshalling problem in only one bay. The steps of the heuristic are explained and illustrated with a numerical example in Figs from 2 to 11, where the maximum height of a stack is 4 containers.

- Step 1: Check the priorities of the containers placed in the first tier and select a container of highest priority. If there is more than a container with the same high priority, then start with the container that the summation of the priorities of the containers above it is higher than of the other container as shown in fig. 2. The selected container will be labeled as the goal container (e.g. the goal container will be shaded with gray in the figures) and the containers above it will be labeled as moving containers.
- Step 2: Move the moving containers to other stacks with top containers having a priority equal to or lower than the priority of the moving container (e.g. if a moving container has a priority of 2, then it will be moved to a stack has a top container of priority either 2 or 3). In this concern two cases may exist; the first one, if the height of the stacks is maximum (e.g. maximum height is 4 containers), while the second case, if the moving container has a priority lower that the priorities of the top containers in all stacks. In both cases a new stack will be constructed as shown in fig. 3. Only one extra stack could be constructed and the containers in the extra stack will be stacked in order of arrival.
- Step 3: Move the goal container to another stack that has a top container with a priority equal to or lower than the priority of the goal container as shown in fig. 4.
- Step 4: Check the priorities of the containers of the first tier, if all the containers have a lower priorities than of the containers above them, then, check the second tier and repeat steps 2, 3 and 4 till all tiers are considered.
- Step 5: In order to relocate the containers in the new constructed stack, the containers in the new stack are moved to other stacks as shown in Fig. 8. This will be done by checking the priorities of the containers that will be moved. First, the moving container will be moved to a stack has a top container has a priority equal to or lower than that of the moving one. If not, then check if there is any empty stack, If there is an empty stack, then construct a new stack with this moving container. If not, then check the priorities of the containers placed below the top containers of the stacks. If one of these containers has priorities equal to or lower than the priority of the moving container, then move this container to any other stack and place the moving container and then replace the container that moved earlier as shown in Figs. 9, 10, and 11.

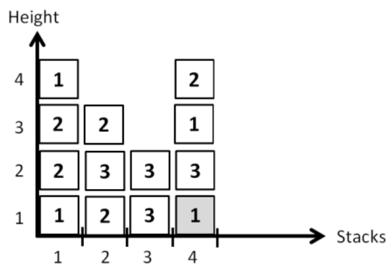

Fig. 2. The initial layout and selection of the goal container

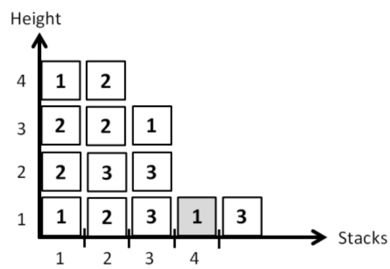

Fig. 3. Layout after moving the moving containers

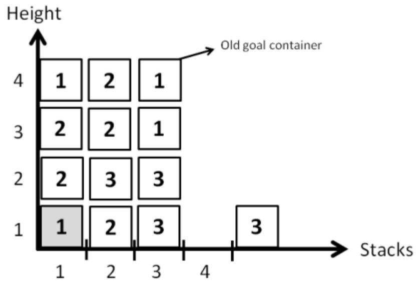

Fig. 4. Layout after moving the old goal container and selecting a new goal container

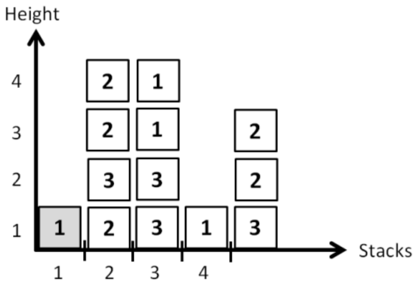

Fig. 5. Layout after moving the moving containers

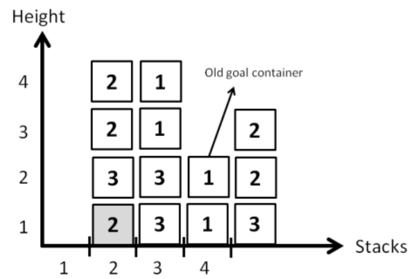

Fig. 6. Layout after moving the old goal container and selecting a new goal container

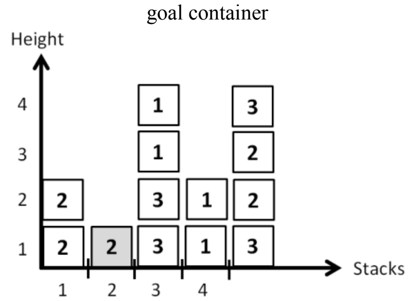

Fig. 7. Layout after moving the moving containers

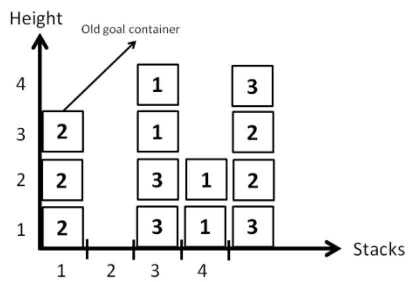

Fig. 8. Layout after moving the old goal container

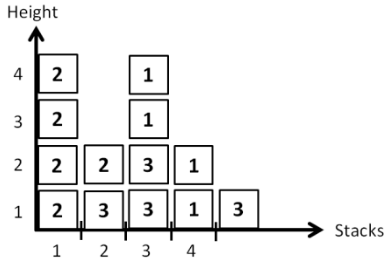

Fig. 9. Layout after moving the extra stack (phase 1)

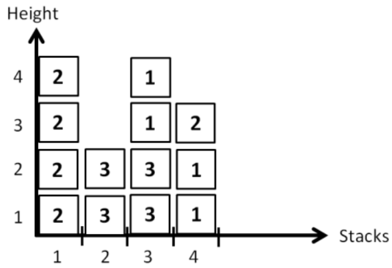

Fig. 10. Layout after moving the extra stack (phase 2)

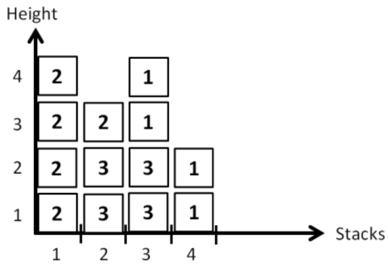

Fig. 11. Final layout

V. ANALYSIS AND RESULTS

In order to measure the performance of the proposed heuristic, the number of movements made during the conversion from the initial layout into the desired final layout is used. The proposed heuristic is to be compared with the work done by Lee and Hsu [7]. They provide an optimization model to solve small problem instances and a heuristic to solve large problem instance. Two cases were selected to make the comparison, in the first case, the total number of containers are 14 containers grouped into three types, and the maximum height of each stack is 4 tiers, while in the second case the total number of containers are 45 grouped into 6 types, and the maximum height of each stack is 5 tiers. Table 1 illustrates the comparison between the results of out proposed heuristic and the results got by Lee and Hsu.

TABLE 1
PERFORMACE OF THE PROPOSED HEURISTIC

	Case 1	Case 2
Number of containers	14	45
The maximum height of a stack	4 tiers	5 tiers
Types of containers	3 types	6 types
Optimum solution	9 movements	______
Proposed heuristic	14 movements	42 movements
Lee and Hsu Heuristic	10 movements	47 movements

In future work, the performance of the heuristic will be measured in terms of the required number of movements and the computation time. These measures would be compared in two ways: (i) The first one is to compare the results with benchmarks done in the field of pre-marshalling problem. (ii) The second one is to use a performance measure to check the quality of the solution. Berthing time is considered as the most important measure for terminal performance as illustrated in [20]

VI. CONCLUSION

The container pre-marshalling problem is one of the important problems existing in container terminals. The literature studying this type of problem is very limited. This paper provided a review for this problem. The proposed heuristic demonstrated initial better performance than the one by Lee and Hsu [7]. The paper also described a heuristic based framework to solve this problem. Concerning the future work, it is planned to perform a comprehensive performance assessment and implement the proposed heuristic in a real world case study and try to investigate how to integrate this problem with other problems in the container terminals.

ACKNOWLEDGMENT

This research project is sponsored by the Egyptian Ministry of higher education grant and the Japanese International Cooperation Agency (JICA) in the scope of the Egypt Japan University of Science and Technology.

REFERENCES

[1] R. Stahlbock, and S. VoB, "Operations research at container terminals: A literature update," *OR spectrum*, vol. 30, pp. 1-52, 2008

[2] W. NG, and K. Mak, " Yard crane scheduling in port container terminals," *Applied Mathematical Modeling*, vol. 29, pp. 263-275, 2005

[3] S. Huang, and T. Lin, " Heuristic algorithm for container pre-marshalling problems," *Computers & Industrial Engineering*, vol. 62, pp. 13-20, 2012

[4] M. Caserta, S. Schwarze, and S. VoB, " A mathematical formulation and complexity considerations for the blocks relocations problem," *European Journal of Operational Research*, vol. 219, pp. 96-104, 2012

[5] A. Bortfeldt, " A heuristic for the container pre-marshalling problems," *In Proceedings of the 3rd International Conference on Computer and IT Applications in Maritime Industries*, pp. 419-429, 2004

[6] A. Bortfeldt, and F. Foster, " A tree search procedure for the container pre-marshalling problem," *European Journal of Operational Research*, vol. 217, pp. 531-540, 2012

[7] Y. Lee, and N. Hsu, " An optimization model for the container pre-marshalling problem," *Computer & Operations Research*, vol. 34, pp. 3295-3313, 2007

[8] Y. Lee, and S. Chao, " A neighborhood search heuristic for pre-marshalling export containers," *European Journal Of Operational Research*, vol. 196, pp. 468-574, 2009

[9] Y. Lee, and Y. Lee, "A heuristic for retrieving containers from a yard," *Computer & Operations Research*, vol. 37, pp. 1139-1147, 2010

[10] M. Caseta, and S. VoB, " A corridor method-based algorithm for the pre-marshalling problem," *Lecture Notes in Computer Science*, vol. 5484, pp. 788-797, 2009

[11] M. Caserta, S. VoB, and M. Sniedovich, " Applying the corridor method to a blocks relocation problem," *OR Spectrum*, vol. 33, pp. 915-929, 2011

[12] KH. Kim, and GP. Hong, " A heuristic rule for relocating blocks," *Computers & Operations Research*, vol. 33, pp. 940-954, 2006

[13] M. Molins, M. Salido, F. Barber, " Intelligent planning for allocating containers in maritime terminals, " *Expert Systems with Applications*, vol. 39, pp. 978-989, 2012

[14] M. Molins, M. Salido, and F. Barber, " Domain-dependent planning heuristics for locating containers in maritime terminals," *IEA/AIE*, pp. 742-751, 2010

[15] M. Salido, M. Molins, and F. Barber, " A decision support system for managing combinatorial problems in container terminals," *Knowledge-Based Systems*, vol. 29, pp. 63-74, 2012

[16] C. Exposito-Izoquierdo, N. Melian-Batista, and M. Morena-Vaga, " Pre-marshalling problem" Heuristic solution method and instance generator," *Expert Systems with Applications*, vol. 39, pp. 8337-8349, 2012

[17] S. Sauri, and E. Martin, " Space allocating strategies for improving import yard performance at marine terminals," T*ransportation Research Part E*, vol. 47, pp. 1038-1057, 2012

[18] T. Park, R. Choe, Y. Kim, and K. Ryu, " Dyanamic adjustment of container stacking policy in an automated container terminal," *International journal of Production Economics*, vol. 133, pp. 385-392, 2011

[19] D. Hou, and F. Chen, " Constraint satisfactory technology for stacking problem with ordered constraints, " *Procedia Engineering*, vol. 29, pp. 3317-3321, 2012

[20] A. Imai, E. Nishimura, and S. Papadimitriou, " Berthing ships at a multi-user container terminal with limited quay capacity," *Transportation Research Part E*, vol. 44, pp. 136-151, 2008