WHAT IS BETTER: 4 TIERS OR 5 TIERS IN THE CONTAINER STACKING PROBLEM?

Miguel A. Salido, Oscar Sapena, Federico Barber

Instituto de Automática e Informática Industrial
Universidad Politécnica de Valencia
Valencia, Spain

msalido@dsic.upv.es, osapena@dsic.upv.es, fbarber@dsic.upv.es

ABSTRACT
Nowadays, there exists a large competition between maritime ports, so that the improvement of customer service became a serious important problem within port container terminals which led to several sub-problems (Vis and Koster, 2003). One of these sub-problems is the Container Stacking Problem. A container stack is a type of temporary store where containers await further transport by truck, train or vessel. The main efficiency problem for an individual stack is to ensure easy access to containers at the expected time of transfer. Since stacks are 'last-in, first-out', and the cranes used to relocate containers within the stack are heavily used, the stacks must be maintained in a state that minimizes on-demand relocations. In this paper, we study and compare the configuration of yard-bays with 4 tiers and yard-bays with 5 tiers.

Keywords: Container stacking problem, Planning, Artificial Intelligence

1. INTRODUCTION

Loading and offloading containers on the stack is performed by cranes. In order to access a container which is not at the top of its pile, those above it must be relocated. This reduces the productivity of the cranes.

Maximizing the efficiency of this process leads to several requirements. First, each incoming container should be allocated a place in the stack which should be free and supported at the time of arrival. Second, each outgoing container should be easily accessible, and preferably close to its unloading position, at the time of its departure. In addition, the stability of the stack puts certain limits on, for example, differences in heights in adjacent areas, the placement of empty and 'half' containers and so on.

Since the allocation of positions to containers is currently done more or less manually, this has convinced us that it should be possible to achieve significant improvements of lead times, storage utilization and throughput using improved techniques of the type indicated.

Figure 1 shows a container yard. A yard consists of several blocks, and each block consists of 20-30 yard-bays (Kim, Park and Ryu, 2000). Each yard-bay contains several (usually 6) rows. Each row has a maximum allowed tier (usually tier 4 or tier 5 for full containers). Figure 2 shows a transfer crane that is able to move a container within a stacking area or moving to another location on the terminal.

Figure 1: A container yard (courtesy of Hi-Tech Solutions)

Figure 2: A Rubber-tired Gantry crane (courtesy of Kalmar Industries)

When an outside truck delivers an outbound container to a yard, a transfer crane picks it up and stacks it in a yard-bay. During the ship loading operation, a transfer crane picks up the container and transfers it to a truck that delivers it to a quay crane.

In container terminals, the loading operation for export containers is carefully pre-planned by load planners. For load planning, a containership agent usually transfers a load profile (an outline of a load plan) to terminal operating company several days before a ship's arrival. The load profile specifies only the container group, which is identified by container type (full or empty), port of destination, and size to be stowed in each particular ship cell. Since a ship cell can be filled with any container from its assigned group, the handling effort in the marshalling yard can be made easier by optimally sequencing export containers in the yard for the loading operation. The output of this decision-making is called the "load sequence list". In order to have an efficient load sequence, storage layout of export containers must have a good configuration.

The container stacking problem is known to be an intractable highly combinatorial optimization problem (Kefi et al., 2008). It is NP-complete so heuristic techniques are necessary to manage this kind of problems. Few studies are dealing with this problem. In (Kim and Bae, 1998) the authors proposed dynamic programming to attain an ideal configuration while minimizing the number of containers to move and the follow-on travelled distance. The problem is divided into two sub-problems: Bay matching and move planning problem, and moving tasks sequencing problem. A mathematical model is used for the resolution of each subproblem. In (Kozan and Preston, 1999) a genetic algorithm was used in their study to reduce container transfer and handling times and thereafter the berthing time of ship at port quays. In (Kim, Park and Ryu, 2000) the authors dealt with the problem of determination of the best storage slot for containers with the aim of minimizing the number of expected relocation movements in loading operation. They deployed the weight of the container as criteria to define certain priority between the different containers to be stacked in the storage yard.

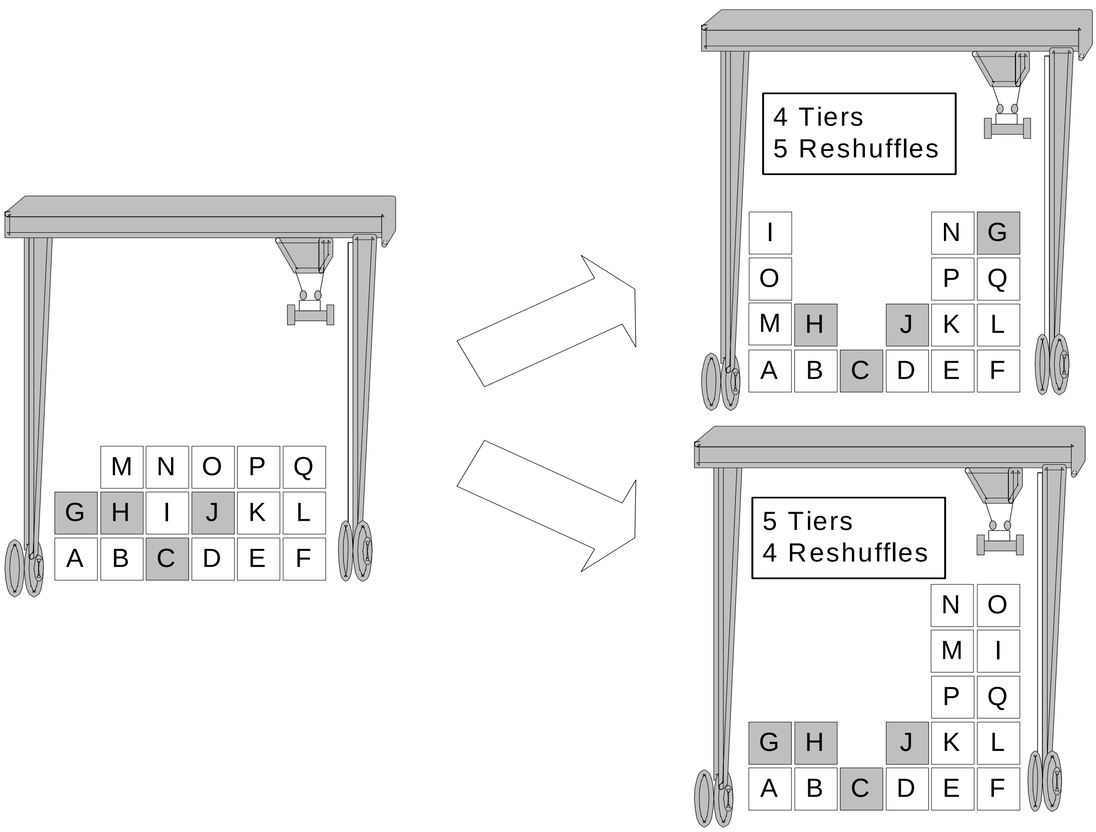

Figure 3: A container yard with 4 and 5 tiers

2. WHAT IS BETTER: FOUR OR FIVE TIERS? A COMPARATIVE STUDY

The main focus of this paper is to give response to harbor operator requirements regarding the best configuration of a yard-bay to minimize the number of reshuffles of export containers. We are interesting on comparing the number of reshuffles in yard-bays with 4 tiers against yard-bays with 5 tiers.

Thus, given a layout, the user selects the set of containers that will be moved to the vessel and the allowed tier. Our tool is able to organize the layout in order to allocate these containers at the top of the stacks in order to minimize the number of relocations. Thus a solution of our problem is a layout where all outgoing containers can be available without carrying out any reshuffle.

We have analyzed two configurations of yards: with 4 tiers and with 5 tiers. We have evaluated the minimum number of reshuffles needed to allocate all selected containers at the top of the stacks or under another selected containers in such a way that no reshuffles is needed to load outgoing containers.

Figure 3 shows an example of a bay with 6 stacks and 4 tiers. The grey containers (G,H,C,J) were selected as outgoing containers. The minimum number of reshuffles to achieve our goal with the restriction of 4 tiers is five. However, the minimum number of reshuffles to achieve our goal with 5 tiers is four. It can be observed that both solutions allow the yard crane to pick up all selected containers without any unnecessary reshuffle.

To evaluate the behavior of both configurations, the experiments were performed on random instances. A random instance is characterized by the tuple <*n*,*s*>, where *n* is the number of containers and *s* is the number of selected containers. Each instance is a random configuration of all containers distributed along the six stack with 4 or 5 tiers. We evaluated 100 test cases for each type of problem.

In Figure 4, we evaluated the number of reshuffles needed for problems <*15,s*> with 4 tiers and 5 tiers. Thus, we fixed the number of containers to 15 and we increased the number of selected containers *s* from 1 to 13. It can be observed that as the number of selected containers increased, the number of reshuffles increased until a threshold in which the number of reshuffles decreases due to the fact that the number of selected containers is closer to the number of containers (a Gaussian curve). The number of reshuffles was similar in both configuration for "easy" problems, that is, with few selected container or many selected containers. However, the number of reshuffles in problems <*15,5*> and <*15,7*> was lower in yard-bays with 5 tiers than in yard-bays with 4 tiers. In both cases, the upper bound was reached when the number of selected containers was 7.

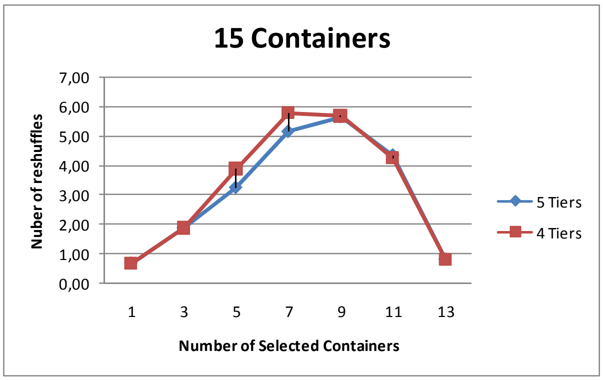

Figure 4: Number of reshuffles for problems <*15,s*> with 4 tiers and 5 tiers.

In Figure 5, we evaluated the number of reshuffles needed for problems <*20,s*> with 4 tiers and 5 tiers. Thus, we fixed the number of containers to 20 and we increased the number of selected containers *s* from 3 to 19. As in the previous figure, it can be observed the Gaussian curve. Nevertheless, due to the tightest of the configuration, mainly for 4 tiers, there were a high number of unsolvable problems with 4 tiers, due to the fact that our tool returns "time out" after 300 seconds. We assigned an upper bound of 22 reshuffles for unsolvable problems. Thus, it can be observed that in problems <*20,13*>, most of the problems were unsolvable for 4 tiers. In all cases, the configuration of 5 tiers maintained a better behavior than the configurations of 4 tiers.

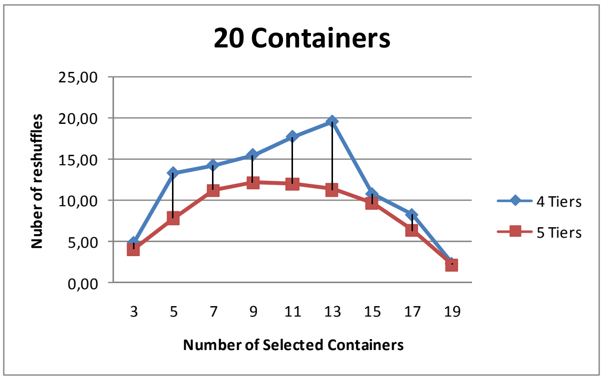

Figure 5: Number of reshuffles for problems <*20,s*> with 4 tiers and 5 tiers.

In Figure 6, we evaluated the number of reshuffles needed for problems <*n,4*>. Thus, we fixed the number of selected containers to 4 and we increased the number of containers n from 11 to 23. It can be observed that as the number of containers increased, the number of reshuffles increased. For small number of containers (low values of *n*) there is no difference between 4 tiers and 5 tiers. This is due to the fact that it is not needed the use of the higher stacks to achieve a solution because there exist many combinations to achieve a solution. However, the number of reshuffles with 5 tiers was lower than the number of reshuffles with 4 tiers for higher number of containers. Due to the fact that we consider yard-bays of 6 stacks, for problems with 4 tiers the maximum number containers is bounded to 24. In this case instances *<23,4>* for problems with 4 tiers generally has no solution. Thus, we can conclude that for low loaded yard-bays (<15 containers) there is not different between 4 tiers and 5 tiers, meanwhile for high loaded yard-bays 5 tiers is more appropriate for minimizing the number of reshuffles.

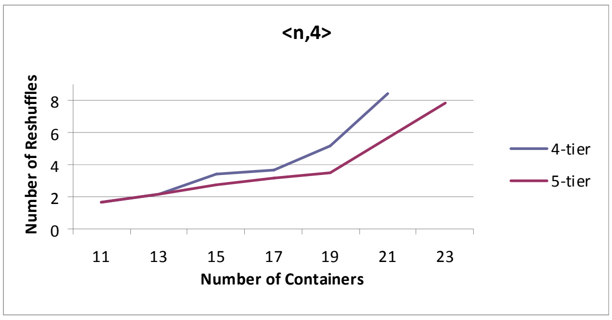

Figure 6: Number of reshuffles for problems *<n,4>* with 4 and 5 tiers.

3. CONCLUSIONS

In this paper we study and compare the number of reshuffles in yard-bays with 4 tiers against yard-bays with 5 tiers. The obtained results recommend the use of stack with 5 tiers due to the fact that the number of reshuffles is reduced for outgoing containers. In further works, we will evaluate these configurations taken into account the distance between the source and the destination of each reshuffle with the aim of minimizing the number of reshuffles and the total distance.

ACKNOWLEDGMENTS

This work has been partially supported by the research projects TIN2007-67943-C02-01 (Min. de Educacion y Ciencia, Spain-FEDER) , P19/08 (Min. de Fomento, Spain-FEDER) and by the Technical University of Valencia.

REFERENCES

Vis, I.F.A. and Koster, R.D., 2003 Transshipment of containers at a container terminal: an overview. In *European Journal of Operational Research*, 147, 1-16

Kim, K.H., Park, Y.M. and Ryu, K.K., 2000. Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124, 89–101

Kefi, M., Korbaa, O., Ghedira, K. and Yim, P., 2008 Heuristic-based Model for Container Stacking Problem. In *19th International Conference on Production Research.*

Kim, K.H., Bae, J.W., 1998. Re-marshaling export containers in port container terminals. *C&IE*, 35, 655-658

Kozan, E., Preston, P., 1999, Genetic algorithms to schedule container transfers at multimodal terminals. *In International Transactions in Operational Research*, 6, 311-329

AUTHORS BIOGRAPHY

Miguel A. Salido is an associate professor in Computer Science at the Technical University of Valencia, Spain. Most of his research is focused on techniques for constraint satisfaction techniques and railway scheduling problems. He is the recipient of some national and international awards. He is author of more than 70 papers published on international journals and conferences. He is editor of some books and guess editor of some international journals. He is member of several Scientific and Organizing Committees.

Oscar Sapena is an associate professor in Computer Science at the Technical University of Valencia, Spain. His research is focused on Planning form the Artificial Intelligence point of view. He is author of more than 20 papers published on international journals and conferences. He is member of several Scientific and Organizing Committees.

Federico Barber is Full Professor with the Department of Computer Science, where he leads a research team in artificial intelligence. He has worked on the development of temporal reasoning systems, Constraint Satisfaction Problems, planning and scheduling. He is the author of about 90 research articles which have been published in several journals and conferences. His research has produced several tools for solving real-world optimization combinatory problems. He has participated in and led several national and European research projects related to these areas. He is currently president of the Spanish Association for Artificial Intelligence, and member of several scientific associations.