In: Proc. 6th International Industrial Simulation Conference, ISC 2008, pp 293-298, June 9-11, 2008 Lyon, France

USING CONTAINER CALL TIME INFORMATION FOR RESTACKING REDUCTION

Wing Sum Lee, Jaap A. Ottjes , Hans P.M. Veeke and Joan C. Rijsenbrij
Delft University of Technology
Faculty of Mechanical, Maritime and Materials Engineering
Transport and Logistic Technology Group,
Mekelweg 2, 2628 CD Delft The Netherlands.
E-mail:J.A.Ottjes@ tudelft.nl

KEY WORDS

container, stacking, discrete event simulation, call time information

SUMMARY

Eight heuristic stacking methods for containers are tested using simulation in a realistic setting of a container stacking lane. As the main performance indicator the 'performance loss factor' is used, defined as the number of restacking moves divided by the number of retrieved containers. Special attention has been paid to the possibility of utilizing uncertain call time information of (a part of) the containers to be processed. Depending on the percentage of containers with call time information, the uncertainty of this information and the filling rate of the stack the reduction of the performance loss factor is determined. Especially methods that evaluate the 'remaining stacking capacity' of all ground slots will yield reductions of the performance loss factor up to 85%.

INTRODUCTION

In deep sea container port operations the main issue is to serve the large container carriers within the contract period. A typical turn around time of carriers is 24 hours. After arrival and berthing of the vessel along the quay, the import containers are unloaded and stacked in the terminal main stack. In general this stack is situated near the deep sea quay wall. The export containers, available in (another part of) the main stack are then loaded into the vessel. After that the vessel is ready to depart.

The container main stack serves as a buffer between deep sea ship and other transport modes. The main stack of a container terminal requires quite a large area because it has to accommodate both export loads and import loads of large container carriers. A container stack consists of a number of ground slots on which containers are stacked in piles with a certain maximum height. Depending of the type of stacking equipment used, a stack may be composed of a number of "modules" each with own stacking/retrieving equipment, for example a portal type stacking crane. Each module has three dimensions: width, length and (maximum) height, see Figure 1.

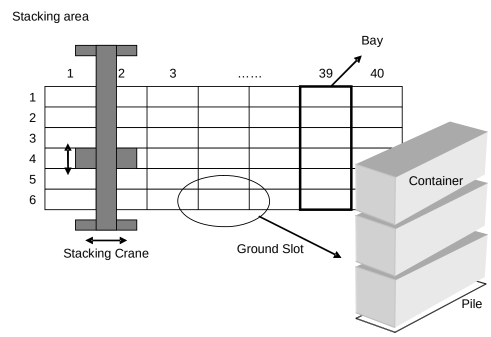

Figure 1 stacking module as used in the simulation experiments with 6 x 40 = 240 ground slots

The stacked import containers have different final destinations and retrieval times. The retrieval time of a container is the time it has to be retrieved from the stack for further transportation. After having been called, for example if a truck has arrived to pick up a specific container, this container has to be retrieved from the stack. If the pile with the container is stacked randomly and is n containers high, the probability that the right one is on top is only 1/n. In the other cases one or more containers of the pile have to be removed and put somewhere else in order to have access to the desired container. We call this "restacking". If during the stacking of an import batch, knowledge of the retrieval time of the containers is taken into account, the number of restacking operations can likely be reduced.

Kim (1997) provides a method to estimate the number of restacking actions to pick up an arbitrary container. Also the total amount of restacking actions to pick up all containers in a bay can be estimated by this method. In later research (Kim and Hong 2006), two restacking methods, branch-and-bound and a heuristic rule, are compared to estimate an expected number of additional restacks for a stack. Narasimhan and Palekar (2002) proved the problem is NP-Complete and also investigated both a branch-and-bound method and a heuristic solution. These articles all consider a stacking area which is divided in bays. Exchange of containers is only done within one bay. Duinkerken, Evers et al. (2001) developed a simulation model of a quay transport system to determine optimal stacking heights, numbers of AGVs and other variables. A stacking method based on categories was developed and tested. Category stacking is already applied at various terminals operated by straddle carriers, rubber tired and rail mounted gantry cranes.

Research Question

The research question in this work is to determine, for a number of (re)stacking methods, the benefit of a relationship between extra call time information of individual import containers and possible reduction of restacking effort.

MODELING APPROACH

The scope of work has been restricted to one stacking module with typical dimensions for a deep sea terminal. The dimensions of the test module are set to the ones shown in Figure 1. Width: 6 containers length: 40 containers and a maximum stacking height of 4 containers. All containers are considered to have the same dimensions. The stack will be filled at time=0 up to a predefined filling rate. The arrival order of the containers is random. The containers are put into the stack according the stacking method applied. After its dwell time, the container is retrieved from the stack and the number of restacks is recorded. After the first fill, the filling rate of the stack is maintained during the simulation run by replenishing the retrieved containers. These new containers are also stacked according the stacking method applied.

All containers are collected within 10 days after arrival. As a rule of thumb, taken from experience of some Rotterdam based terminals, we assume that 60% of the total is collected the first 4 days and 40% the remaining 6 days. This gives a mean time in stack (dwell time) of 4 days. The dwell time pattern is shown in Figure 2.

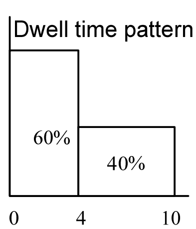

Figure 2 simplified dwell time distribution (days)

Each container is assigned an “estimated retrieval time” (ERT) and a confidence range of b time units. b is a parameter in the experiments. The estimated retrieval time is used in the (re)stacking algorithms. The estimated retrieval time is sampled for each container using the dwell time distribution of Figure 3. It is used in some of the stacking algorithms. The real retrieval time (RRT) is used to generate the actual retrieve action. In the model the real retrieval time of each container is sampled from a uniform distribution at (ERT- b and ERT + b). We distinguish containers with extra call time information and containers without extra call time information. For containers without call time information the ERT is not used for (re)stacking purposes.

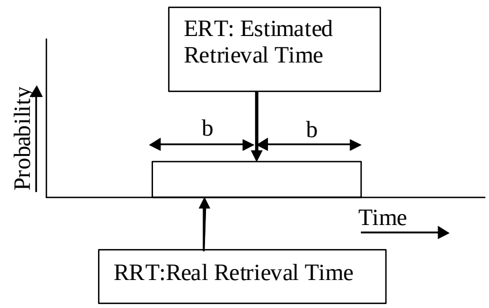

Figure 3 Estimated Retrieval Time (ERT), Real Retrieval Time (RRT) and time window (-b,+b) around the ERT in which RRT falls.

STACKING METHODS

Eight different stacking methods are tested. In some methods use is made of the expected retrieve time of containers. If a fraction, say x%, of the containers has an ERT and consequently 100-x% have not, the available slots are divided proportionally.

If a container has to be (re)stacked, a “cost” factor, determined according the actual (re)stacking method, is assigned to each ground slot. The slot with the lowest cost factor will be chosen as the destination of the container to be (re)stacked. If multiple slots have the lowest cost factor the nearest position with respect to the origin of the container to be handled is selected. In some stacking methods the time axis is split up into a number of discrete time intervals of length c hours. Each container is then assigned the serial number of the time interval corresponding with its ERT. Next all eight methods will be discussed.

Method 1: Random

The method “random” does not use the ERT. For (re)stacking the target slot is randomly sampled from all slots that are not full.

Method 2: Leveling

The method “leveling” does not use the ERT. It takes the pile height of each slot as the cost factor.

In Table 1 an example of the each of the remaining methods 3 through 8 is shown. A square represents a container. The number in the square indicates the container’s ERT. After the table each method is explained.

In: Proc. 6th International Industrial Simulation Conference, ISC 2008, pp 293-298, June 9-11, 2008 Lyon, France

Table 1 example of stacking methods 3 through 8. 10 time windows are available.

5	Container to be (re)stacked. 5 means ERT = 5

Example of a Stack. Max pile height: 4. The numbers in the Boxes indicate the ERT	7			
	2	4		
	1	8	3	
Pile Height	3	2	1	0
3:*Minimum Time Difference*				
Cost factor	2	-1	-2	10
Priorities	***1***	3	4	2
4:*Remaining Stack Capacity (RSC)*				
Cost factor	2	-2	-6	40
Priorities	***1***	2	3	4
5:*RSC Improved*				
Cost factor	2	-2	-6	40
Priorities	***1***	2	3	4
6:*RSC Improved with continuous Time*				
Cost factor	2	-2	-6	40
Priorities	***1***	2	3	4
7:*RSC Pile Search*				
Cost factor	-12	-2	-6	40
Priorities	4	2	3	***1***
8:*Restack Chance*				
Cost factor	5	2	2	1
Priorities	4	2	2	***1***

Method 3: Minimum Time Difference

The method "Minimum Time Difference" uses ERT. The cost factor of a slot is the difference of the ERT of its top container and the ERT of the container to be stacked. If the slot is empty, the cost factor is set to the maximum possible time step (10 time units in the examples of Table 1). If no slots with positive time difference are found, in a second loop the slot with the least negative minimum time difference is selected. If there are still no choices available a third loop is performed. This third loop will search through slots assigned to containers with no ERT information.

Method 4: Remaining Stack Capacity (RSC)

In the method "RSC" the ERT is projected on the time axis that is divided in time steps of length **c** time units (Duinkerken et al 2001). The index (n) of a step indicates the ERT. The cost factor of a slot is defined as:

*RSC = (maximum possible height – height of the top container) * (* $n_{topcontainer} - n_{container\text{-}to\text{-}be\text{-}stacked}$ *).*

If a slot is empty, the maximum possible time window will be used. In the example of Table 1 the maximum time window is 10 time units. After calculation of the RSC values, the slot with the smallest RSC >= 0 is preferred. If no slot is available, the slot with the least negative RSC value is selected. If there are still no choices available, a third loop is performed. This third loop will search through slots that are actually destined for containers without ERT information.

Method 5: RSC Improved

The only difference with method 4 is that now, after calculation of the RSC values, the slot with the smallest RSC >0 is preferred. An extra demand is that $(n_{topcontainer} - n_{container\text{-}to\text{-}be\text{-}stacked}) > 0$. In Table 1 this leads to the same answer as method 4, however in case the container to be stacked would have ERT=7, the empty slot will get preference.

Method 6: RSC Improved with continuous time axis.

This method is the same as method 5, RSC Improved except the partitioning of the time axis. Instead of discrete time windows, it uses a continuous time axis for RSC calculations.

Method 7: RSC Pile Search

This method is derived from method 5, RSC Improved. Instead of only looking at the top container of a slot, the container **in** the slot that gives the lowest RSC is normative for the score of the corresponding slot. RSC values lower than zero are taken into account too. Looking at Table 1, the cost factor of the first slot referring to the bottom container becomes:
RSC= (4-1)*(1-5) =-12 and for the next slot:
RSC= (4-2)*(4-5) = -2.

Table 2 Example of cost factor calculation for method 8.

Pile with ERT's	#restacks
5	0
7	0
2	2
1	3
Cost Factor for table 1	5

In: Proc. 6th International Industrial Simulation Conference, ISC 2008, pp 293-298, June 9-11, 2008 Lyon, France

Method 8: Expected Restacks

This method uses a measure for the expected number of restacks per slot after the container to be stacked would have been placed on top. For each container in a pile the number of restacks is determined assuming that this particular container is called according its ERT. The numbers are added up for all containers in the pile, forming the cost factor of that pile. An empty slot gets a restack number of 1 to promote the usage of available stacks that do not require restacks.
In Table 2, the score is calculated for the first slot of the example of Table 1: after placing the new container on top, retrieving each container individually in the pile will cause a total number of restacks of 5.

EXPERIMENTAL RESULTS

In Figure 4 a part of the simulation screen is shown. The model is build in 'TOMAS' and is configurable for stacking lanes of any dimension. The eight stacking methods can be selected for each run. It's possible to make a distinction between restacking method and stacking method. Further all varied parameters can be adjusted. The container dwell time distribution can be selected from predefined distributions or entered manually.

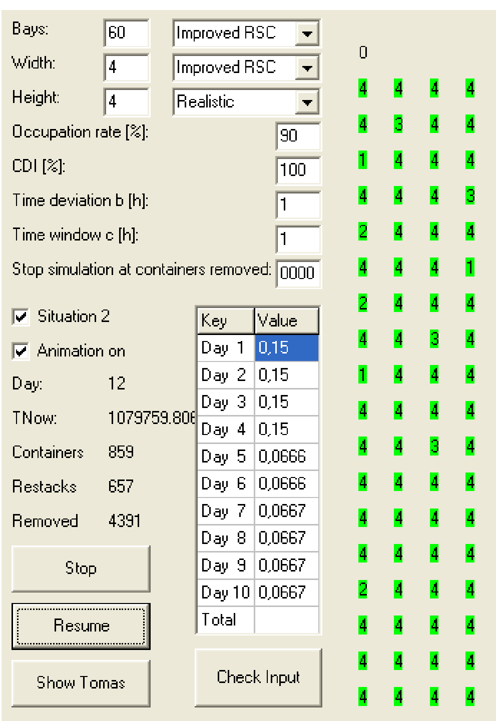

Figure 4 screen shot of the model

The model has been applied on a stack module with typical dimensions, according Figure 1. The tests are performed always for all stacking methods. The parameters varied for the consecutive sets of runs are:

- Run time, expressed in numbers of removed containers. Default value: 100.000 containers
- Percentage of Container with Departing Information: CDI factor (in %). Default value: 100%
- The pursued average Occupation rate of the stack (in %). Default value: 90%
- The possible deviation of the real retrieving time (=estimated retrieving time ± b). Default: b=1 h.
- Time step in case the time axis is split up in discrete time intervals of length c (hours). Default: c=1 h.

Figure 5 shows the performance loss factor for different run lengths. It shows that all outcomes stabilize from about a run length corresponding with 10000 removed containers. All runs reported are performed with a run length corresponding with 100000 removed containers. The relative standard deviation of the Performance Loss Factor proved to be less than 5%. It is concluded that both methods 5 and 6 (RSC improved and RSC Improved with continuous time axis) outperform the other methods. The number of restacks is reduced with a factor 15 compared to random stacking.

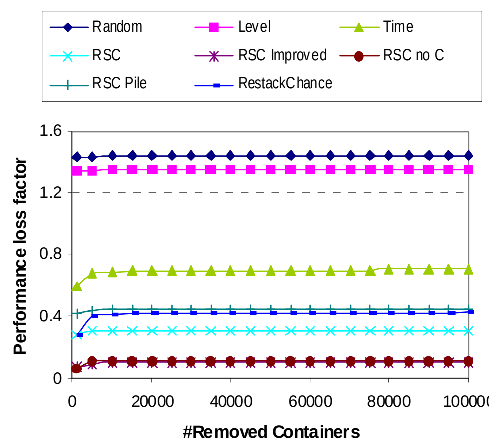

Figure 5 Performance loss factor for different run lengths CDI = 100%, Occupation rate = 90%, b = 1 hr, c = 1 h.

Figure 6 shows the results of a set of runs varying the percentage of containers with call time information. All methods, except of course those not using ERT information, perform better if more information is available. It can be concluded that, in case of 50% containers with information, the number of restacks can be halved.

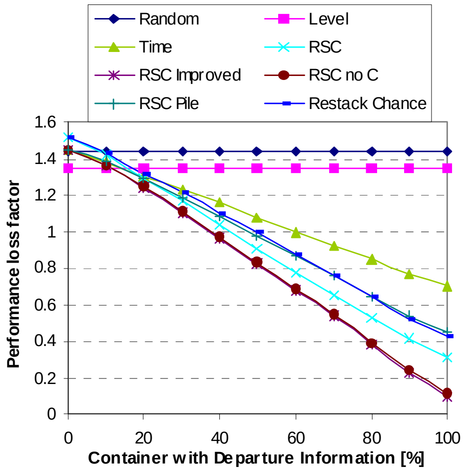

Figure 6 Performance loss factor vs. varying % of containers with departure information CDI = 0-100%, Occupation rate = 90%, b = 1 hr, c = 1 hr

In Figure 7 the influence of the average stack filling rate is analyzed. It can be expected that at low occupation and thus low piles, restacking will be reduced. Still it is salient that both improved RSC methods hold on giving a large reduction in restacking until 90% filling rate. At filling rates close to 100%, the Performance Loss Factor becomes 1.5 for the first two methods. This corresponds to the value obtained theoretically, assuming infinite number of slots and 100% occupation and a maximum height of 4 containers.

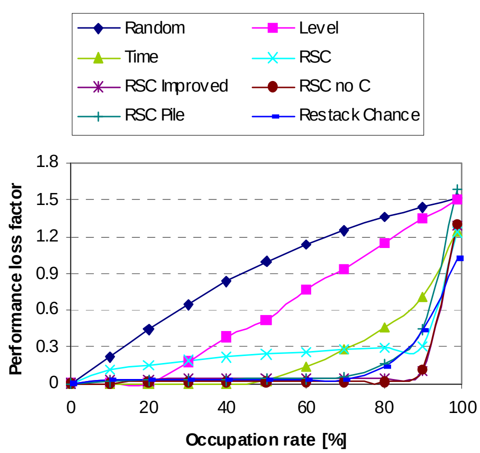

Figure 7 Performance loss factor vs. varying average stack occupation [CDI = 100%, Occupation rate = 0-99%, b = 1 hr, c = 1 hr]

In the last set of runs the accuracy of the estimated call time is varied by varying the value of b. Increasing b will introduce larger deviations between the real call time and the estimated one. One would expect the performance to be worse with increasing b. That expectation is confirmed in Figure 8. Still it is concluded that all methods using ERT give a reduction in the Performance Loss Factor of at least 50 % in case of an uncertainty of RRT of ± 6 h. Method 5, RSC improved even shows a reduction of 70% in that case.

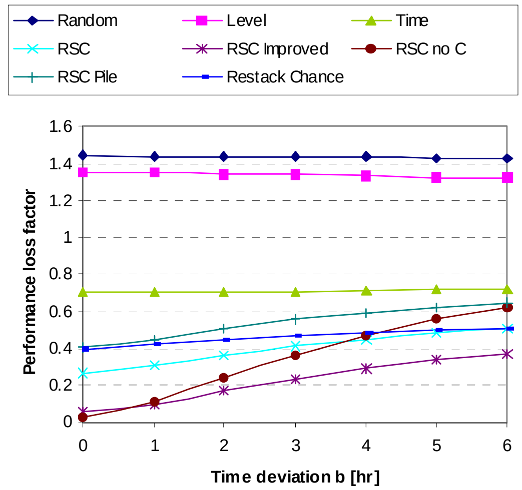

Figure 8 Performance loss factor vs. b, indicating the accuracy of the estimated call time. CDI = 100%, Occupation rate = 90%, b = 0-6 hr, c = 1 hr

CONCLUSIONS

In this work the results are shown of simulation experiments with eight heuristic (re)stacking methods to be used in container terminal operation.

If use is made of incomplete and, to certain extend, unreliable information of retrieval times of containers in (re)stacking operations, the number of restacking activities can be reduced considerably. Compared to the methods not using information (method 1 and 2), stacking methods using extra information perform significantly better. Methods using “improved minimal remaining stacking capacity” as “cost” criterion perform best. Some quantitative conclusions are:

- The number of restacks can be reduced with 50% if for half of import containers the retrieval times are know with a possible deviation of ± 1 hour and the stack average filling rate is 90%.
- When for all containers the departing time information is known within a bandwidth of 2 hours, the number of restacks in the considered case can be reduced with a factor 15.
- If the uncertainty of the real retrieving time in relation to the estimated retrieval time is ± 6 hours, all methods using departing information give a reduction in the Performance Loss Factor of at least 50 %. Method 5, RSC improved even shows a reduction of 70% in that case.

In general the simulation results clearly indicate that pre-information of container retrieval times can be very helpful to diminish retrieval times and thus to improve service times.

FUTURE WORK

The model is only applied on a reference case of one stacking module with fixed dimensions. The model will be further employed by investigating the influence of other interesting parameters like call patterns, call batches for example for loading a train or barge. The model is ready for realistic simulation as a part of a container terminal model with a multi-module stack. In that case the cycle times of stacking cranes and assignment of containers to stacking lanes will be taken into account as well as developing an extension for serving land side modalities.

REFERENCES

Duinkerken, M.B., J.J.M. Evers, J.A. Ottjes (2001). *A Simulation Model for Integrating Quay Transport and Stacking Policies on Automated Container Terminals*. Proc. of the 15th European Simulation Multiconference (ESM2001). June 2001. Prague [SCS].

Kim,K.H. and Gyu-Pyo Hong (2006). *A heuristic rule for relocating blocks.* Computers & Operations Research 33 (2006) 940–954

Kim, K.H. (1997). *Evaluation of the Number of Rehandles in Container Yards*. Computers & Industrial Engineering, vol. 32, no. 4, pp. 701-711.

Narasimhan, A and Palekar, (2002) Analysis and Algorithms for the Transtainer Routing Problem in Container Port Operations Transportation Science © 2002 INFORMS Vol. 36, No. 1, February 2002 pp. 63–78

Tomas 2007. Web site: www.tomasweb.com

Veeke, Hans P.M., Jaap A. Ottjes, 2000. *Tomas: Tool for Object-oriented Modeling And Simulation*. In proceedings of Advanced Simulation Technology Conference (ASTC2000). April 16-20, 2000, Washington, D.C. pp. 76-81. The Society for Computer Simulation International (SCS), ISBN: 1-56555-199-0