

Energy Efficient Container Yard Stacking

Alexandre Goussiatiner
Container Terminal and Transportation Specialist
Modern Port Technologies Inc.
E-mail: info@modernport.com
Web: www.modernport.com

Introduction

Dramatic increases in fuel and electricity costs as well as new requirements to reduce engine emissions have put additional pressure on most container terminal operators worldwide. In response to this challenge a number of manufacturers of container handling cranes started to concentrate on designing a new generation of equipment employing a variety of energy regeneration schemes. In our opinion, this effort must be matched by significant improvements in the operational planning and control areas. The reality is that even coming close to 100% crane energy efficiency will not cardinally change the situation if the operational inefficiencies remain.

The most significant causes of the operational inefficiency are as follows:

- Large number of unproductive moves (reshuffling) during retrieval from the imports. Statistics from large marine container terminals indicate that delivery of 100 import containers require more than 100 reshuffling moves on average and the percentage of the unproductive moves increases with stack height. To a certain degree, implementation of the appointment reservation system can help as it provides time slots for the import container delivery and allows preplanning for the import stack. But the effect of the appointment reservation system is limited since most of the containers do not have reservations at the time when they are unloaded from the vessel and stored in the yard.
- Inefficient handling of heavy containers. Energy required for hoisting and trolley traveling depends on the container weight. Not taking into account the weight leads to the situation when ‘heavy’ lifts are traveled the same distances and heights as ‘light’ lifts.

At the same time, computing power of the industrial computers and servers has been significantly increased during the last few years. This fact provides incentives to improve energy efficiency using software components which can help to make energy efficient container handling decisions.

To investigate the benefits of different stacking strategies, Modern Port Technologies Inc. has developed **CS** - container stacking software component. **CS** is designed to make ‘low level’ decisions for RTG/RMG operators. It advises where to put incoming containers and plans sequences of the moves when reshuffling is required for retrieval. The ‘horizon’ for the decision-making is one yard section (bay). This low ‘horizon’ of the decision making allows **CS** to complement various Terminal Operating Systems (TOS) which are making decisions on higher levels.

The target of **CS** is to reduce power input required for the operation in the long run. As it achieves the goal, the following operational benefits should be expected:

- Reducing energy cost per container handled
- Reducing engine emissions (of the diesel powered cranes)
- Increasing crane gross productivity
- Lowering maintenance and repair costs since crane working conditions are improved: cranes perform fewer working cycles in a month and the portion of the time when crane handles heavy loads is reduced.
- Reducing wait time for the road trucks and yard prime movers as the number of unproductive moves is reduced.
- Potentially increasing in the storage capacity as operators will be able to increase stack height.

Reinforcement Learning - “Brain” of the Stacking Component

Each time **CS** makes a decision it seeks to maximize the *accumulated reward* over time and uses *policy* which connects the decision with the reward. As *policy* gets updated over time, **CS** ‘learns’ how to act from consequences of its actions in the past. Updates of the *policy* are governed by algorithms developed in the Reinforcement Learning (part of the modern Machine Learning Theory).

As the *accumulated reward* indicator **CS** uses the average power spent for a container leaving a section over a long time period of time (i.e. one month). Usually during its lifetime in the yard (three to seven days average) a container gets handled multiple times: during initial stacking, reshuffling and retrieval. Each handling requires power input for hoisting, trolley traveling and lowering.

CS uses its own operational model of the yard and collects historical data on the movements. For instance, for the initial stacking, **CS** records hoisting height and takes into account the height of the stack the crane must get over to move the container, trolley traveling distance to reach the column and lowering height (see Fig.1).

To estimate power input required for the moves, CS uses a simplified analytical model which approximates the power required for each element of the move. The power input required for the hoisting/lowering a container is calculated using mechanical efficiency of the hoisting drive, weight of the spreader, hoisting/lowering height. Similarly, power input required for the trolley traveling is calculated based on mechanical efficiency of the trolley drive, coefficient of rolling friction, total weight to be moved including weight of the container, spreader and trolley and horizontal distance.

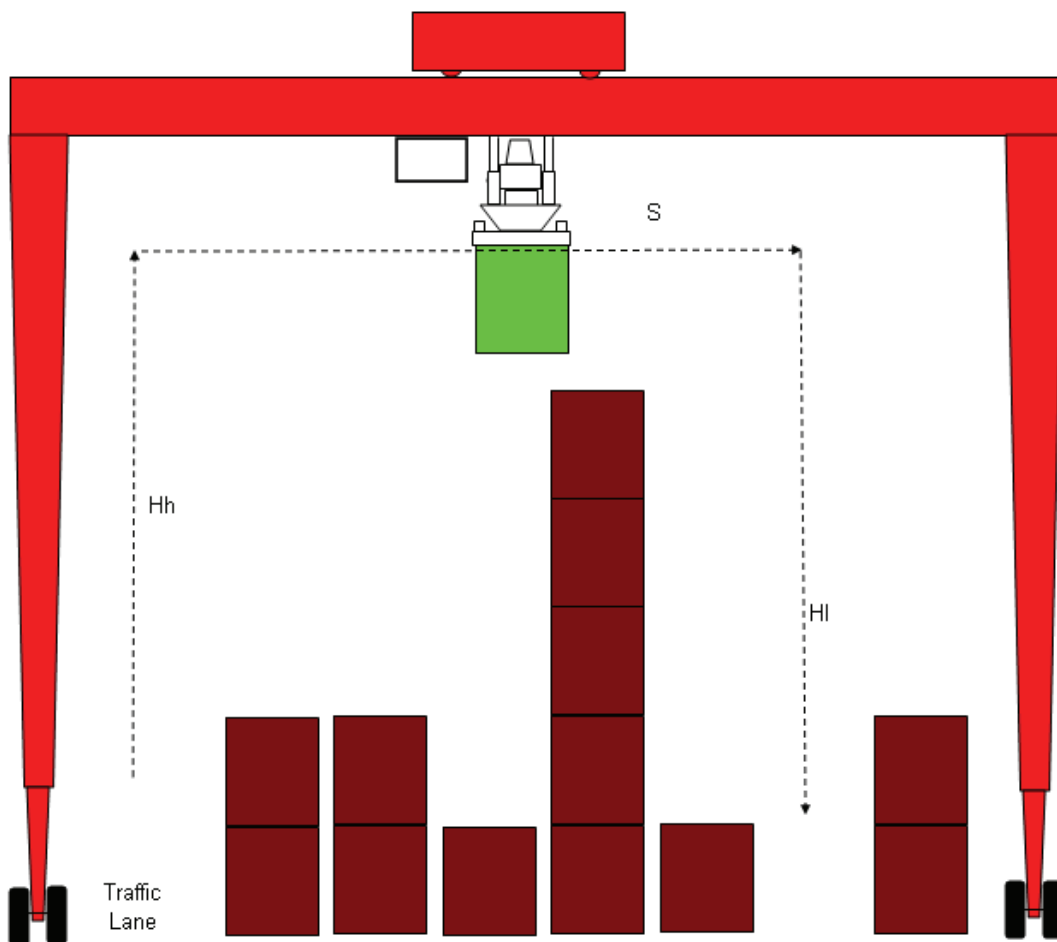


Figure 1. Rubber-Tired Gantry Crane

The **policy** determines all allowed choices and evaluates each of them. The nature of the task is such that the number of variants is relatively small: for the incoming container the number of variants is equal to number of columns available for stacking and for reshuffling planning, the maximum number of variants is lower than $n^{(m-1)}$ where n is the number of columns in the section and m is maximum height of the stack.

Assume that the **policy** evaluates a move to place a container in a column.

Following are some of the parameters the **policy** takes into account when it evaluates the immediate results of the move:

- Column Number - sequential number for the column (starting from the traffic lane)
- Height - height of the container in the column
- Power Input - the amount of power input required for hoisting, trolley traveling and lowering the container to the column;
- Reshuffling Probability - probability of the event that the container is reshuffled.

In addition to the evaluation of the immediate results of the move, **policy** also uses parameters describing the condition of the entire section after the potential move is done, such as:

- Column Heights – number of containers in each column starting from the closest one from the traffic lane
- Reshuffling Probability Index - number of containers in each column which have Reshuffling Probability greater than certain level (i.e.0.5)

Reshuffling Probability

To operate, **CS** has to be able to reliably predict that a container will be reshuffled if placed on a certain column. Suppose container C_t is located on top of the column with four others: $\{C_1, C_2, C_3, C_4\}$. Reshuffling Probability - probability of the event R that C_t is reshuffled can be calculated as following:

$$P(R) = 1 - ((1 - P(C_t, C_1)) * (1 - P(C_t, C_2)) * (1 - P(C_t, C_3)) * (1 - P(C_t, C_4)))$$

Where $P (C_t , C_1)$ -probability of the event that container C_1 is retrieved before Ct

$P (C_t , C_2)$ -probability of the event that container C_2 is retrieved before Ct

Suppose we have pair of containers {Ca, Cb} with the “ages” (number of days in the terminal) $\eta_a = 1$ and $\eta_b = 4$. The following is the description of the algorithm used in CS to estimate probability P(Ca,Cb):

If we take into account the container line operator and consignee as the factors, we can retrieve historical data for the containers with identical characteristics and create following sampling distributions for each container (Fig.2):

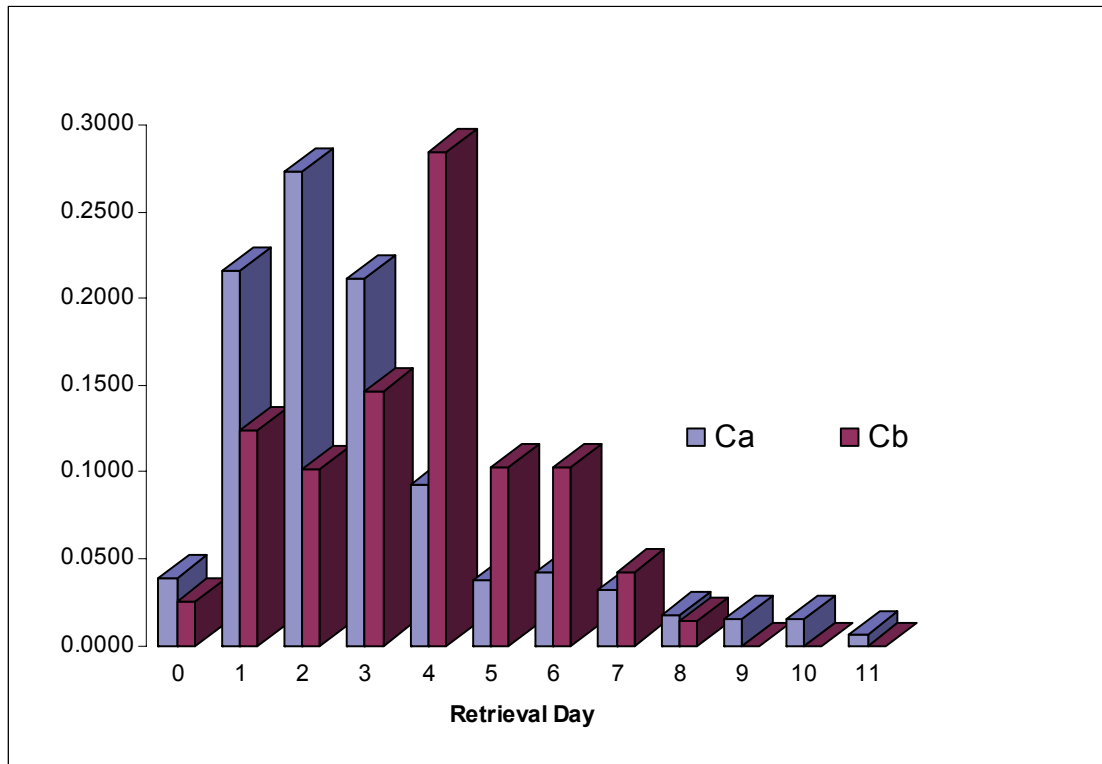


Figure 2. Sampling Distributions

The next step would be to make can make a probability multiplication table (Fig.3) and determine probability using the following formula:

$$P (C_a , C_b) = \frac{X + 0.5 * Y}{Z}$$

Where Z - Probability of the event that the first container has an age greater or equal to η_a and second container has an age greater or equal to η_b (total of red, yellow and green cells).

X - Probability of the event that the first container is retrieved on a day prior to the day when a second container is retrieved (total of red cells).

Y - Probability of the event that the first container is retrieved on the same day as the second one (total of yellow cells).

		0	1	2	3	4	5	6	7	8
		0.0253	0.1240	0.1013	0.1467	0.2847	0.1031	0.1031	0.0419	0.0140
0	0.0391	0.00099	0.00485	0.00396	0.00574	0.01113	0.00403	0.00403	0.00164	0.00055
1	0.2162	0.00548	0.02682	0.02191	0.03173	0.06157	0.02229	0.02229	0.00907	0.00302
2	0.2737	0.00693	0.03394	0.02773	0.04016	0.07792	0.02821	0.02821	0.01147	0.00382
3	0.2120	0.00537	0.02630	0.02148	0.03111	0.06037	0.02185	0.02185	0.00889	0.00296
4	0.0926	0.00235	0.01149	0.00938	0.01359	0.02637	0.00955	0.00955	0.00388	0.00129
5	0.0379	0.00096	0.00470	0.00384	0.00556	0.01079	0.00391	0.00391	0.00159	0.00053
6	0.0424	0.00107	0.00526	0.00430	0.00622	0.01207	0.00437	0.00437	0.00178	0.00059
7	0.0319	0.00081	0.00395	0.00323	0.00468	0.00908	0.00329	0.00329	0.00134	0.00045
8	0.0174	0.00044	0.00216	0.00177	0.00256	0.00497	0.00180	0.00180	0.00073	0.00024
9	0.0153	0.00039	0.00190	0.00155	0.00225	0.00437	0.00158	0.00158	0.00064	0.00021
10	0.0153	0.00039	0.00190	0.00155	0.00225	0.00437	0.00158	0.00158	0.00064	0.00021
11	0.0060	0.00015	0.00075	0.00061	0.00088	0.00171	0.00062	0.00062	0.00025	0.00008

Figure 3. Multiplication of probabilities table

For example: if $\eta_a = 1$ and $\eta_b = 4$, we will get

$$P(C_a, C_b) = \frac{0.229 + 0.5 * 0.143}{0.666} = 0.452$$

The above algorithm was tested in the simulation environment. CS was presented with a large number (1,000,000) of container pairs and was requested to provide probability of the event that second container is retrieved before first one. If the probability was greater than or equal to 0.5, it was assumed that it is predicted that Cb will be retrieved before Ca. The prediction was compared with historical data. For the pairs with the age difference greater than a certain threshold, analyses revealed that the percentage of the correct answers increases if the probability is set to one. To summarize, percentage of the correct answers varied from 62% - zero age difference to 99% - 11 days age difference (Fig.4).

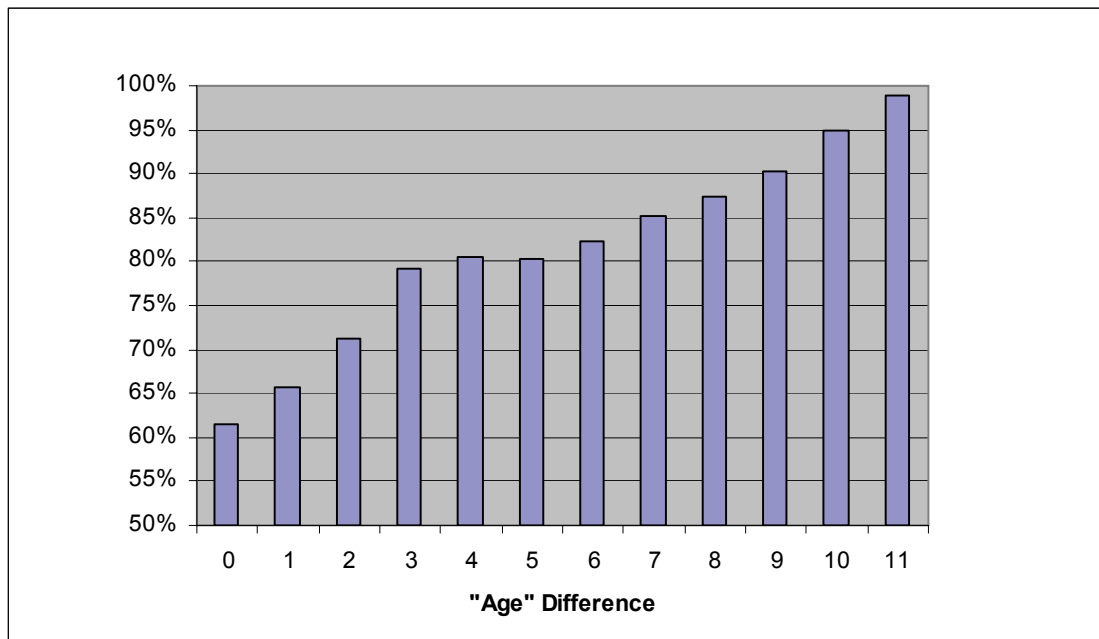


Figure 4. Percentage of the correct prediction