Container Relocation Problem

SOFTWARE SUMMARY

January 16, 2018

Purpose

Due to limited space within the storage yard, containers are stacked in columns that are only accessible from the top. Thus, in order to retrieve a container that is not on the top of a stack, the blocking containers above must be relocated to a different column. Deciding where to relocate the blocking container can have a strong impact on the efficiency of the storage yard, as optimal placement can minimize the number of future relocations needed, reducing wait time for customers, operating costs for port managers, and environmental impact.

Currently, relocation moves are decided by the planner and are based on moving the blocking container to a nearby column of the shortest height. The proposed **Container Relocation Problem** (**CRP**) software will automatically determine the optimal relocation move for blocking containers on retrieval request, as well as placement of new containers within a block. The process will use expected container retrieval order to maximize yard efficiency by minimizing the number of future relocations that are caused by the placement of blocking or new containers.

Functionality

The **CRP** software has two functions:

Case 1. Determining placement within a block for relocation of a blocking container during retrieval of a requested container.

Case 2. Determining placement within a block when stacking a new container

At a high level, the **CRP** software takes in a container ID to retrieve (**case 1**) or a block location to stack a new container (**case 2**) and outputs the optimized set of moves that should be taken to complete this action.

Software Module

The **CRP** software acts as a smart-stacking decision module which communicates with the **Terminal Operating System** (**TOS**). Thus, the software will be referred to as **CRPM** (**Container Relocation Problem [Decision] Module**).

CRPM receives data *from* the **TOS** about containers currently stacked in the yard to determine departure order, as well as sending the list of optimal moves *to* the **TOS** as instructions for use by crane operators. The module will continuously refresh its information and update container expected retrieval order in real-time based on changes in the yard. If the terminal has implemented a **Prediction System** (**PS**) for container retrieval times or **Truck Appointment System** (**TAS**), **CRPM** will also receive data from these.

Components

The module consists of four components:

1. **Departure Order Generator:** Using the **TOS** data (and **PS** and **TAS** data, if available), generates an order of departure for all containers in the terminal. The data is all updated in real-time, adapting to short-notice requests.
2. **Virtual Port:** Stores data for all containers in the port, such as location (block, row, column, position) and retrieval order. Receives data from the **TOS** and **Departure Order Generator**. The **Virtual Port** is updated in real-time.
3. **Stacking Move Generator:** Using the departure order from the **Virtual Port** and a placement block chosen by the planner, outputs the best column to stack a container as an operating instruction to the **TOS**.
4. **Retrieval Relocation Move Generator:** Using the departure order data from the **Virtual Port** and the container ID to retrieve, generates a list of moves resulting in the retrieval of the desired container and outputs this to the **TOS** as a list of operating instructions.

Inputs

CRPM communicates in real-time with the **TOS** (and **PS** and **TAS**, if existent) to maintain a store of current information about the terminal yard. This picture of the yard, stored in the **Virtual Port** is supplemented with departure order information from the **Departure Order Generator,** and then forms the base informational input the optimization algorithm needs to run—container locations and order of departure.

For both **CRPM** functional cases, retrieval and stack, specific user input is required in addition to the automatically processed yard information. For the retrieval case, the planner will enter the container ID of the desired container to be retrieved. For the new container stacking case, the operator must enter the block in which the new container will be stacked.

To summarize, CRPM takes as input:

Container locations
Container departure order
Container ID (case 1) or Block ID (case 2)

Outputs

In both functional cases, **CRPM** returns a list of operating instructions which will be sent to the **TOS**. For the stacking of a new container, the list will contain a single instruction. For container retrieval, the last instruction in the list will be the retrieval of the desired container.