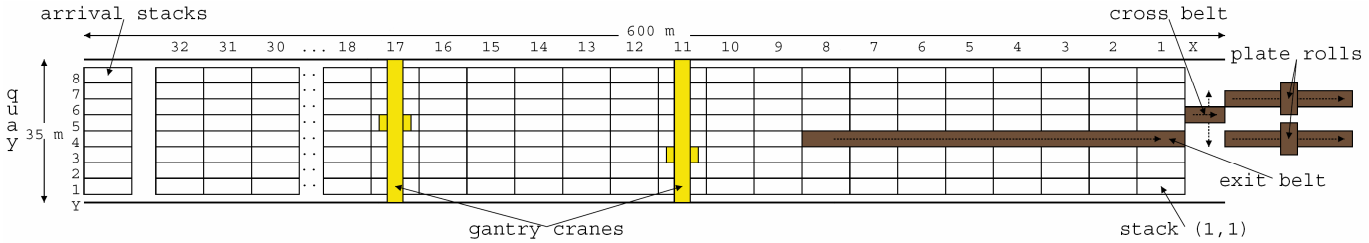
Optimizing the Steel Plate Storage Yard Crane Scheduling Problem

Using a Two Stage Planning/Scheduling Approach



Problem description

The problem at hand is from a large steel shipyard. Steel ships are constructed by welding together large pre-cut steel plates. The raw plates arrive to the storage, where they are stored until they are used in the production. The plates are so large and heavy that the only possible way of storing them is by placing them in stacks. The plates are moved by electromagnets mounted on two gantry cranes.

For all plates, we have a due date, i.e. the time where this plate is to be moved to the exit belt. For all plates with due date equal to the day of planning, we further have an ordering in which they must be put on the exit belt.

The figure shows the storage yard as seen from above. Each square represents a stack of steel plates. Using the two gantry cranes, all due plates must be moved to the exit belt. The due plates may be located beneath other plates and hence other plates may have to be moved first.

The task is to generate a feasible and cost efficient schedule for the cranes. The cranes move on the same set of tracks and as a consequence the two cranes can never pass each other.

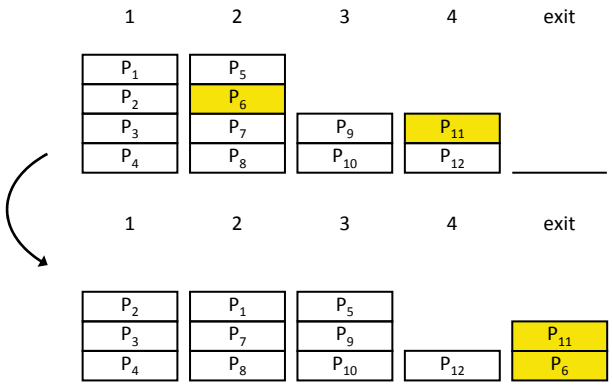
Modeling the Problem

The problem is modeled in two stages: a planning stage and a scheduling stage.

In the planning stage, coordination of the cranes is ignored to facilitate faster optimization. The plan consists of a number of consecutive moves taking us from the current state of the yard to a final state of the day. In the final state, all plates with deadline on the current day are brought to the exit belt. At the same time, the plan should leave the yard in the best possible condition for the next day.

In the scheduling stage a crane is allocated to each move and each move is given a specific time of execution.

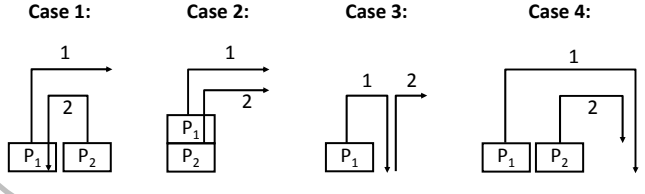
Toy example in one dimension with five stacks and two due plates



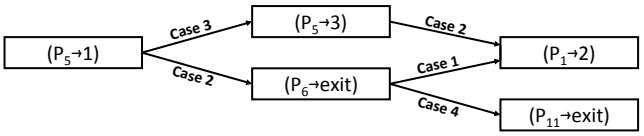
As can be seen in the figure, the exit belt is modeled as a special stack. Given the storage state of the top figure, a schedule must be generated, where the two due plates P_6 and P_{11} are moved to the exit belt (in that order). A possible result of the planning stage is the following sequence of moves, yielding the final state shown in the lower of the two figures above. **Result of planning stage:**

(P_5→1)	(P_6→exit)	(P_5→3)	(P_{11}→exit)	(P_1→2)

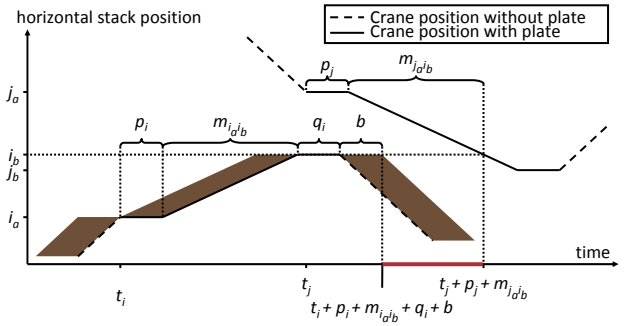
The sequence of moves is turned into a fully descriptive schedule. First, we need to analyze dependencies between the moves. There are four cases where the order of the moves has to be respected to guarantee that the final state is unchanged. If two moves fit into any of these four cases, Move 1 must be scheduled before Move 2. Move 1 and Move 2 are depicted by the two arrows in the figure.



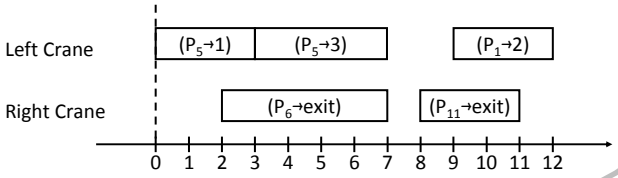
Presented below are the relationships between the moves of this example. This means, for example, that the order of moves (P_5→3) and (P_6→exit) is irrelevant to the final state and they may be reordered or carried out in parallel.



To avoid crane collision, for each pair of moves, we calculate a required gap between the initiations of the two moves. The gap may be negative or even non-existing if the moves are mutually independent. Below is an example of such a calculation for two moves i (from stack i_a to stack i_b) and j (j_a to j_b). The calculation includes initiation time t, lift time p, move time m, drop time q, and safety buffer between cranes b (b is the width of the brown area). To avoid collision (and respect the safety buffer), the size of the red time interval may never decrease below 0.



In a very simple world, where moving from one stack to its neighbor takes 1 time unit and lifting or dropping a plate takes 1 time unit and when no additional buffer is used, the schedule below is feasible in the example. **Result of scheduling stage:**



Anders Dohn - adh@imm.dtu.dk

Technical University of Denmark