

# A note on a model for quay crane scheduling with non-crossing constraints

Alberto Santini

*Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy.*  
a.santini@unibo.it

Henrik Alsing Friberg

*MOSEK Aps, Denmark.*  
haf@mosek.com

Stefan Ropke

*Department of Management Engineering, Technical University of Denmark, Denmark.*  
ropke@dtu.dk

## Abstract

This article studies the Quay Crane Scheduling Problem with non-crossing constraints, which is an operational problem that arises in container terminals. An enhancement to a mixed integer programming model for the problem is proposed and a new class of valid inequalities is introduced. Computational results show the effectiveness of these enhancements in solving the problem to optimality.

## 1 Introduction

A container terminal manager is faced with several interesting and challenging optimization problems and the topic of applying operational research methods to optimize container terminal operations has received a great amount of attention in recent years. The most important container terminal optimization problems as well as related solution methods are surveyed by [6] and [5].

The focus of this article is on the quay crane scheduling problem (QCSP). In the QCSP a container vessel and a number of quay cranes are given and the objective is to make a schedule for the quay cranes such that the tasks that need to be performed on the vessel are carried out in a way that satisfies both the terminal manager and the vessel owner. Typically it is of primary importance to serve the vessel as quickly as possible. This is in the interest of the terminal manager, as it ensures that valuable quay space is freed up quickly and that labor cost is kept in check. It is also in the interest of the vessel owner, because it means that the ship can quickly commence its voyage, so to minimize unproductive time.

A conceptual container vessel is displayed in Figure 1. The figure shows that storage space on the vessel is divided into bays, rows and tiers, with a certain bay–row–tier combination pointing out a cell in the vessel that can store one forty feet container. This figure is, of course, a simplification. In practice the containers are not stored in a box-shaped vessel, the system for numbering positions on the vessel is different from what is used here and containers come in different sizes. The reader is referred to, for example, [4] for a more realistic description of a container vessel. For the purposes of this work, the simple description is sufficient since, as it is common in the QCSP literature, the assumption is made that each task consists of unloading and loading an entire bay.

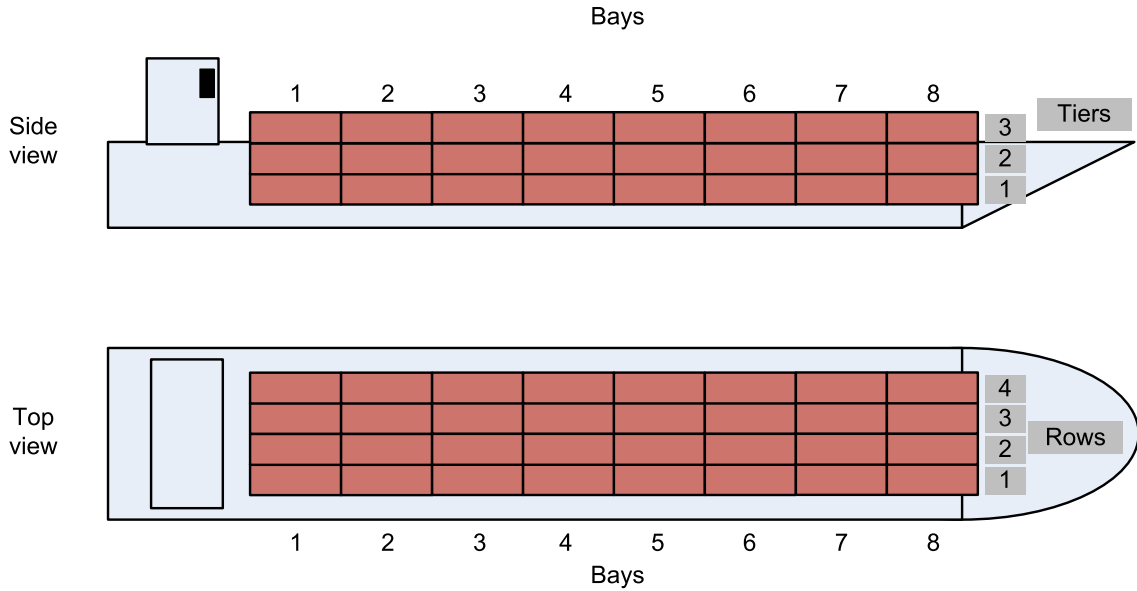


Figure 1: Conceptual container vessel

The QCSP model studied in this article is the one presented in [3] and the contribution of the article is to show how the model, in a very simple way, can be improved to make it much more tractable for off-the-shelf solvers like CPLEX. Let  $B = \{1, \dots, n\}$  be the set of bays,  $K = \{1, \dots, m\}$  the set of quay cranes and  $p_b$  the processing time of bay  $b \in B$ . Each crane can process one bay at a time. Once the processing has started it has to run to its end. Cranes are running on rails, so they cannot overtake each other. The dimensions of bays and cranes are such that it is impossible to place two or more cranes at any bay simultaneously. It must be decided which crane should process which bay and at what time, while respecting the non-crossing constraint and the necessary time for processing each bay. It is assumed that the time for moving the crane between bays is negligible compared to the time for processing each bay. The objective is to minimize the make-span of the entire operation; that is, to minimise the ending time for the crane that ends the latest.

A classification scheme for QCSP formulations as well as a survey of contributions to the problem are presented in [1]. QCSP formulations are classified according to four attributes: 1) task attribute, 2) crane attribute, 3) interference attribute and 4) performance attribute. The QCSP studied in this article is classified as “Bay | – | cross | max(compl)” which means that 1) each individual task is a bay — as opposed to a group of bays or a single container at the two extremes, 2) there are no special attributes associated with cranes, 3) the non-crossing of cranes is respected and 4) the maximum completion time of all tasks is minimized.

## 2 Mathematical model

The mathematical model is based on that of [3] which in turn is an improved version of the model presented by [2]. The model uses the binary variable  $x_{bk}$  which is 1 if and only if bay  $b \in B$  is served by crane  $k \in K$ , the binary variable  $y_{bb'}$  is 1 if and only if work on bay  $b \in B$  is finished before work on bay  $b' \in B$  starts. The variables  $c_b$  indicate the completion time of bay  $b \in B$  and  $c$  is the overall makespan. Using these variables and letting  $M$  be a sufficiently large positive integer number, the model is:

$$\min c \tag{1}$$

$$\begin{aligned}
\text{subject to } c &\geq c_b && \forall b \in B && (2) \\
c_b &\geq p_b && \forall b \in B && (3) \\
\sum_{k \in K} x_{bk} &= 1 && \forall b \in B && (4) \\
c_b &\leq c_{b'} - p_{b'} + M(1 - y_{bb'}) && \forall b, b' \in B, b \neq b' && (5) \\
\sum_{k \in K} kx_{bk} - \sum_{k \in K} kx_{b'k} + 1 &\leq M(y_{bb'} + y_{b'b}) && \forall b, b' \in B, b < b' && (6) \\
\sum_{k \in K} kx_{b'k} - \sum_{k \in K} kx_{bk} &\leq b' - b + M(y_{bb'} + y_{b'b}) && \forall b, b' \in B, b < b' && (7) \\
x_{bk} &= 0 && \forall b \in B, k \in K, k > b && (8) \\
x_{bk} &= 0 && \forall b \in B, k \in K, n - b < m - k && (9) \\
x_{bk} &\in \{0, 1\} && \forall b \in B, k \in K && (10) \\
y_{bb'} &\in \{0, 1\} && \forall b, b' \in B, b \neq b' && (11) \\
c_b &\in \mathbb{R} && \forall b \in B && (12) \\
c &\in \mathbb{R} && && (13)
\end{aligned}$$

The objective function (1) minimizes the total make-span of the process. Constraint (2) together with the minimization of the objective function ensures that  $c$  is equal to the largest of all completion times. Constraint (3) makes sure that the completion time of each bay is greater than its processing time. Constraint (4) ensures that every bay is served by exactly one crane. Constraint (5) links the  $y_{bb'}$  and  $c_b$  variables. It forces  $y_{bb'}$  to zero whenever  $c_b > c_{b'} - p_{b'}$ , that is, when  $b'$  is started before  $b$  finishes. Constraint (6) makes sure that the cranes do not cross and that each crane is working at one bay at a time. Constraint (7) ensures that there is always enough space between two cranes (e.g. that crane 1 and 3 never are servicing two adjacent bays simultaneously). Constraints (8) and (9) ensure that no crane is pushed outside the bounds of the ship. This is illustrated in Figure 2 that shows an example with 8 bays and 3 quay cranes. In this example it is only crane 1 that is feasible for bay one; crane 2 and 3 are not feasible since that would imply that crane 1 is pushed further left and there may not be space for that since another vessel may be moored directly to the left of the current vessel or the vessel may be at the end of the quay. Similarly it is only crane 2 and 3 that can serve bay 7 since serving it by crane 1 would imply that crane 3 is pushed out of bounds. In the example, constraint (8) fixes  $x_{12}, x_{13}$  and  $x_{23}$  to zero and thereby ensures that no crane is pushed too far left. Constraint (9) fixes  $x_{71}, x_{81}$  and  $x_{82}$  to zero implying that no crane is pushed too far right.

| Bay number           | 1 | 2      | 3           | 4           | 5           | 6           | 7      | 8 |
|----------------------|---|--------|-------------|-------------|-------------|-------------|--------|---|
| Feasible quay cranes | 1 | 1<br>2 | 1<br>2<br>3 | 1<br>2<br>3 | 1<br>2<br>3 | 1<br>2<br>3 | 2<br>3 | 3 |

Figure 2: Bays and feasible quay cranes

The model is different from that of [3] in two ways. [3] creates two dummy bays and two dummy cranes in order to avoid cranes being pushed out of bounds. The dummy bays are situated at each end of the ship and the dummy cranes are locked to serving the two dummy bays during the entire planning period. As explained earlier, in this model the same issue is handled by the variable fixing done in (8) and (9). This modeling approach is preferred, as it requires fewer decision variables and constraints, while making the model easier to understand as well.

The second difference is that constraint (17) of [3] has been left out. Using the notation of this paper, the constraint is

$$c_b + My_{bb'} \geq c_{b'} - p_{b'} \quad \forall b, b' \in B, b \neq b'$$

It forces  $y_{bb'}$  to 1 when  $c_b < c_{b'} - p_{b'}$ , that is, when  $b'$  starts after  $b$  finishes. Forcing the  $y_{bb'}$  variable to one has no impact on the solution of the model since the only other place where  $y_{bb'}$  occurs is in constraints 6 and 7 and here a value of one implies that the constraint will never be binding. The only drawback is that the  $y_{bb'}$  sometimes can have a value 0 in the final solution when the value logically should be 1, but that is not an issue as the only interest is in the values of the  $x_{bk}$ ,  $c_b$  and  $c$  variables.

The following simple family of valid inequalities has been introduced and its significant impact on computing experiments will be later shown:

$$c \geq \sum_{b \in B} x_{bk} p_b \quad \forall k \in K \quad (14)$$

Inequality (14) simply forces the overall make-span to be greater than the sum of all the processing times of the bays served by the same crane.

### 3 Computational results

The purpose of the computational results is to show the impact of inequality (14) when solving model (1) – (13). The computational tests were performed using a 2.93 GHz Intel Core i7 model 940 that has 4 cores. The MIP model was solved using CPLEX 12.4 which was allowed to use all cores of the computer and was allotted one hour per run. Table 1 shows results on the 24 instances used in [3] and compare results with and without constraint (14), as well as the results reported by [3]. The authors obtained the original data set from Lee and Chen and conducted the experiments using these instances.

The first column in the table reports the instance name, the first number gives the number of bays while the second gives the number of quay cranes. The next 6 columns report results from the mathematical model, including constraint (14). The first three of these columns report the lower and upper bounds when CPLEX terminated and the corresponding gap is calculated as  $(UB-LB)/LB \cdot 100\%$ . The next columns report the time spent by CPLEX, where a dash indicates that the solver timed out. The last two of the six columns report if the problem was solved to optimality and the number of branch and bound nodes explored. The following six columns show the same information for the model without constraint (14). The second to last column reports the best solution found by [3]. Values marked with superscript “A” were found using CPLEX, while values marked with superscript “B” were found using a heuristic. The last column reports if the instance was solved to optimality in [3].

A first observation is that the valid inequality has a tremendous impact on the model. Consider for example the first instance. Without the inequality, CPLEX needs about 90 times as much time and needs to explore around 290 times as many nodes in the branch and bound tree in order to solve it to optimality. CPLEX is able to solve 15 instances to optimality when using the inequality and only 4 instances without the inequality. For the instances that none of the models can solve to optimality, the gap is much lower for the model using the inequalities.

When comparing to the results reported by [3], it can be noticed that even the model without the valid inequality is able to solve more instances to optimality. This has been attributed to the fact that the experiment reported in this paper are using a faster computer and a more recent version

of CPLEX. [3] used a 3 GHz Pentium IV computer and did not report which version of CPLEX they used. The authors do not believe that the fact that they are using slightly fewer variables and constraints in their model has a great impact on CPLEX's ability to solve the problem.

The optimal results obtained with the proposed valid inequalities are often substantially better than the heuristic solutions reported in [3] and for most of the instances that were not solved to optimality, CPLEX is still able to find a better solution than Chen and Lee's heuristic. On the other hand, their heuristic is much faster and never uses more than 15 seconds.

The heuristic is also able to find a better solution than CPLEX for the largest instances with 100 bays. However, no container ship has 100 bays so such an instance is not realistic: one of the largest container ships currently in operation, Emma Maersk, has approximately 23 bays (based on inspection of photos). It is therefore possible to conclude that the enhanced model, within one hour, is able to solve most of the realistic sized instances to optimality.

## 4 Conclusion

In this article the quay crane scheduling model proposed by [3] has been revisited. A simple family of inequalities has been introduced and this has been shown to have a great impact on the ability to solve the model to optimality. Computational results showed that the improved model is able to solve most instances with realistic size to optimality. The authors believe that the model can provide inspiration for further work in this and related areas and that the computational results provided can be used as a basis for comparison for future heuristics for the problem.

## References

- [1] C. Bierwirth and F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202:615–627, 2010.
- [2] D.-H. Lee, H.Q. Wang, and L. Miao. Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E*, 44(1):124–135, 2008.
- [3] Der-Horng Lee and Jiang Hang Chen. An improved approach for quay crane scheduling with non-crossing constraints. *Engineering Optimization*, 42(1): 1–15, 2010. ISSN 0305-215X. doi: 10.1080/03052150902943020. URL <http://www.tandfonline.com/doi/abs/10.1080/03052150902943020>.
- [4] D. Pacino, A. Delgado, R.M. Jensen, and T. Bebbington. Fast generation of near-optimal plans for eco-efficient stowage of large container vessels. *Lecture Notes in Computer Science*, 6971: 286–301, 2011.
- [5] R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30:1–52, 2008.
- [6] D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research – a classification and literature review. *OR Spectrum*, 26:3–49, 2004.

| Instance | With constraints (14) |      |       |          |     | Without constraints (14) |       |      |       |          | Lee & Chen |          |                   |     |
|----------|-----------------------|------|-------|----------|-----|--------------------------|-------|------|-------|----------|------------|----------|-------------------|-----|
|          | LB                    | UB   | Gap % | Time (s) | Opt | BB Nodes                 | LB    | UB   | Gap % | Time (s) | Opt        | BB Nodes | Best Solution     | Opt |
| 16-4     | 726.0                 | 726  | 0.0   | 3.4      | ✓   | 12878                    | 726.0 | 726  | 0.0   | 305.4    | ✓          | 3726726  | 726 <sup>A</sup>  | ✓   |
| 16-5     | 586.0                 | 586  | 0.0   | 1.1      | ✓   | 5288                     | 586.0 | 586  | 0.0   | 8.8      | ✓          | 52274    | 610 <sup>A</sup>  |     |
| 17-4     | 741.0                 | 741  | 0.0   | 34.6     | ✓   | 74285                    | 698.0 | 741  | 6.2   | —        |            | 23593606 | 746 <sup>A</sup>  |     |
| 17-5     | 600.0                 | 600  | 0.0   | 44.9     | ✓   | 76721                    | 600.0 | 600  | 0.0   | 330.5    | ✓          | 3435505  | 604 <sup>A</sup>  |     |
| 18-4     | 720.0                 | 720  | 0.0   | 41.2     | ✓   | 84801                    | 687.0 | 720  | 4.8   | —        |            | 29252698 | 737 <sup>A</sup>  |     |
| 18-5     | 579.0                 | 579  | 0.0   | 35.9     | ✓   | 26854                    | 579.0 | 579  | 0.0   | 233.7    | ✓          | 2587339  | 595 <sup>A</sup>  |     |
| 19-4     | 702.0                 | 702  | 0.0   | 255.0    | ✓   | 317052                   | 578.0 | 702  | 21.5  | —        |            | 14824165 | 711 <sup>B</sup>  |     |
| 19-5     | 567.0                 | 567  | 0.0   | 414.2    | ✓   | 404794                   | 542.0 | 567  | 4.6   | —        |            | 21702526 | 580 <sup>A</sup>  |     |
| 20-4     | 925.0                 | 925  | 0.0   | 579.4    | ✓   | 1043278                  | 679.0 | 925  | 36.2  | —        |            | 9578048  | 949 <sup>A</sup>  |     |
| 20-5     | 739.0                 | 739  | 0.0   | 271.4    | ✓   | 323089                   | 677.0 | 749  | 10.6  | —        |            | 14664665 | 781 <sup>B</sup>  |     |
| 21-4     | 759.0                 | 759  | 0.0   | 1151.1   | ✓   | 1516153                  | 540.1 | 759  | 40.5  | —        |            | 6954439  | 801 <sup>B</sup>  |     |
| 21-5     | 612.0                 | 612  | 0.0   | 3320.3   | ✓   | 3703084                  | 524.0 | 612  | 16.8  | —        |            | 8713794  | 622 <sup>B</sup>  |     |
| 22-4     | 757.0                 | 757  | 0.0   | 1203.2   | ✓   | 2042917                  | 612.0 | 759  | 24.0  | —        |            | 6820790  | 766 <sup>B</sup>  |     |
| 22-5     | 611.0                 | 611  | 0.0   | 2715.0   | ✓   | 6097888                  | 544.0 | 611  | 12.3  | —        |            | 12185027 | 636 <sup>A</sup>  |     |
| 23-4     | 886.0                 | 886  | 0.0   | 1916.2   | ✓   | 2168284                  | 562.0 | 889  | 58.2  | —        |            | 5371125  | 910 <sup>B</sup>  |     |
| 23-5     | 708.8                 | 719  | 1.4   | —        |     | 3148789                  | 546.9 | 713  | 30.4  | —        |            | 5754008  | 740 <sup>B</sup>  |     |
| 24-4     | 857.5                 | 860  | 0.3   | —        |     | 3722243                  | 600.0 | 861  | 43.5  | —        |            | 4996441  | 874 <sup>B</sup>  |     |
| 24-5     | 686.0                 | 698  | 1.7   | —        |     | 5210886                  | 525.7 | 693  | 31.2  | —        |            | 3865710  | 712 <sup>B</sup>  |     |
| 25-4     | 1083.5                | 1087 | 0.3   | —        |     | 2085828                  | 579.0 | 1089 | 88.1  | —        |            | 1020039  | 1129 <sup>B</sup> |     |
| 25-5     | 866.8                 | 871  | 0.5   | —        |     | 2651609                  | 560.1 | 877  | 56.6  | —        |            | 2023905  | 921 <sup>B</sup>  |     |
| 50-8     | 998.3                 | 1025 | 2.7   | —        |     | 357369                   | 427.0 | 1013 | 137.2 | —        |            | 62120    | 1046 <sup>B</sup> |     |
| 50-10    | 798.6                 | 830  | 3.9   | —        |     | 204419                   | 448.0 | 839  | 87.3  | —        |            | 2396552  | 897 <sup>B</sup>  |     |
| 100-8    | 2064.9                | 2132 | 3.2   | —        |     | 319813                   | 365.0 | —    | —     | —        |            | 673507   | 2124 <sup>B</sup> |     |
| 100-10   | 1651.9                | 1757 | 6.4   | —        |     | 275514                   | 345.4 | —    | —     | —        |            | 561873   | 1747 <sup>B</sup> |     |

Table 1: Computational results