The multiport continuous berth allocation problem with speed optimization

B. Martin-Iradi[a,*], D. Pacino[a] and S. Ropke[a]

[a] DTU Management, Technical University of Denmark, Kongens Lyngby, Denmark
bmair@dtu.dk, darpa@dtu.dk, ropke@dtu.dk
[*] Corresponding author

Extended abstract submitted for presentation at the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

April 11, 2022

Keywords: Container terminal, Berth allocation, Maritime logistics, Speed optimization, Matheuristics

1 INTRODUCTION

The ongoing growth in shipping cargo globally is urging terminals and shipping line companies to increase the efficiency and sustainability of their operations. The berth planning of a terminal is categorized as one of the most critical sea-side operations due to the scarcity of berthing space (Steenken *et al.*, 2004). It can be modelled mathematically as the Berth Allocation Problem (BAP) where the aim is to assign incoming ships to berthing positions along the terminal.

When each terminal does its planning independently, a delay in one terminal can potentially be propagated through the shipping service to other ports (Notteboom & Vernimmen, 2009) or incur higher fuel costs for the carriers (shipping lines) if they need to increase the vessel's speed to make up for lost time.

A potential solution to avoid this type of scenarios is to establish some form of collaboration between players in the maritime industry. Collaboration can be established not only between same type of stakeholders (i.e., between multiple carriers) but also between more players (i.e., carriers and terminal operators). This is the goal of the Multiport Berth Allocation Problem (MBAP), first introduced by Venturini *et al.* (2017), which simultaneously plans the berth allocation of multiple ports taking into account the vessels' speed.

Currently, exact methods have successfully been applied to the version of the problem with a discrete quay (Martin-Iradi *et al.*, 2022). In this variant, we divide the quay into a discrete set of berths, with only one ship allowed to one berth at a time. The continuous variant of the MBAP allows ships to berth at any point along the quay as long as a safe distance is kept to other ships. In reality, the mooring bollards are distributed at least 10 meters apart from each other restricting the positions where ships can berth. Therefore, a 10 meter discretization of the quay is assumed to be a fair approximation of the continuous version of the problem.

2 PROBLEM FORMULATION

We present two formulations for the continuous MBAP: (1) a mixed-integer problem formulation and, (2) a generalized set partitioning problem formulation.

The MIP formulation can be seen as an hybrid between the formulation for the continous BAP from Kim & Moon (2003) and the formulation for the discrete MBAP from Venturini *et al.* (2017).

In this study, we focus on the set partitioning formulation where each column represents a feasible schedule for a ship (i.e., sequence of berthing positions and times along the ports visited). Ω is the set of columns and λ_j is a binary variable that defines if column j is part of the solution or not. The cost of column j is denoted by c_j. The parameter A_j^i is 1 if column j corresponds to ship i and 0 otherwise. The parameter $B_j^{p,b,t}$ is 1 if column j occupies berthing position b at time t at port $p \in P$ and 0 otherwise.

$$\min \sum_{j \in \Omega} c_j \lambda_j \tag{1}$$

$$\sum_{j \in \Omega} A_j^i \lambda_j = 1 \quad \forall i \in N \tag{2}$$

$$\sum_{j \in \Omega} Q_j^{p,b,t} \lambda_j \leq 1 \quad \forall p \in P, b \in B_p, t \in [s^p; e^p) \tag{3}$$

$$\lambda_j \in \{0,1\} \quad \forall j \in \Omega \tag{4}$$

The objective function (1) minimizes the cost of the solution columns. In our case, it is a weighted sum of the operational costs for both carriers and port operators, namely waiting, handling and delay time at port and fuel consumption when sailing between ports. Constraints (2) enforces one column in the solution for each ship and constraints (3) ensures that there is no overlapping of berthing periods and positions between ships by at most allowing one ship to be berthing at each position and time instant. Finally constraints(4) define the binary property of the decision variables.

The set of columns Ω can be defined using a graph representation. We define a directed and acyclic graph (DAG) for each ship. The graph is defined on a time-expanded network meaning that each node does not only represent a berthing position at a given terminal but also it has a berthing time associated to it. We therefore restrict the set of nodes to those that correspond to feasible berthing positions and times at each port that the ship visits. Feasibility of a node is defined by the operational constraints (e.g., time windows and processing times) at the port but also the ones that a ship can visit given the range of sailing speeds. An arc in the graph connects two nodes of ports that are visited consecutively by the ship. The time and position of the destination node of the arc is given by the time and position of the source node and the required minimum speed to arrive on time. Based on the premise that a ship will not sail faster than required, there is at most one arc connecting a pair of nodes. The reader is referred to Martin-Iradi *et al.* (2022) for a more detailed description of a similar representation used for the discrete MBAP.

3 SOLUTION METHOD

We solve the relaxed version of (1)-(4) using column generation and use a branching strategy to achieve the optimal integer solution.

A pricing problem per ship is defined using the time-expanded network mentioned in the previous section. The dual values from the master solution can be directly updated on the weight of the graph nodes and, therefore, columns can be efficiently generated solving a shortest path problem using a dynamic programming algorithm.

Finding an efficient branching strategy is important in a *branch-and-price* method. Branching on the master variables or on a single node or arc of the pricing problem can be highly unbalanced, therefore, we use a branching strategy based on a set of nodes that tries to have a significant impact in both child branches.

Given a linear and fractional solution to the problem, we group the berthing time and positions of all the columns for each ship and port visit. Then, we measure the average and standard deviation in berthing times and positions separately for all ships and port visits. Finally, our branching candidate will be the ship and port visit with the highest variance either in position or in time. If the highest variance is in berthing time, the child branches will enforce the ship to berth before or after the average time, whereas if the highest variance is in position, the child branches will enforce the ship to berth to the left or right of the average time. By exploring both the time and space partitions, we always guarantee a branching candidate for this strategy, otherwise the solution is integer.

4 PRELIMINARY COMPUTATIONAL RESULTS

We compare the *branch-and-price* method with the MIP formulation solved by CPLEX. All instances consider 3 ports where the number of ships planned ranges between 13 and 19. The route of each ship contains between 2 and 4 port visits. We use CPLEX v12.10 as a solver and a total time limit of 1 hour. The results are summarized in Table 1.

Table 1 – *Average results on a set of 54 instances*

Method	***Branch-and-price***	**CPLEX**
% of instances where an upper bound is found	44	80
Optimality gap (%)	1.50	4.45

We observe that CPLEX is able to find feasible and better solutions to more instances, whereas the *branch-and-price* method, thanks mostly to a tighter root node relaxation achieves a better optimality gap. Both exact methods scale poorly when the instances become larger. This leads us to explore heuristic or matheuristic methods. Preliminary results of a simple randomize *fix-and-optimize* heuristic show that a better integer solutions can be found for most instances (see Table 2). This comparison is done in a subset of 26 instances where CPLEX is able to find an integer solution within the time limit. We observe that most of the solution improvements of the heuristic often occur within the first 10 minutes. This can be useful from a real-life planning perspective where re-planning needs to be done within a short period.

Table 2 – *Performance of the fix-and-optimize heuristic compared to CPLEX with a 1 hour time limit*

Objective compared to CPLEX	**Worse**	**Equal**	**Better**
Fix-and-optimize	6	8	12

We are currently exploring this research direction further and developing a matheuristic method that takes advantage of the capabilities of both exact methods and scales better to larger real-life instances.

References

Kim, Kap Hwan, & Moon, Kyung Chan. 2003. Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological*, **37**(6), 541–560.

Martin-Iradi, Bernardo, Pacino, Dario, & Ropke, Stefan. 2022. The Multiport Berth Allocation Problem with Speed Optimization: Exact Methods and a Cooperative Game Analysis. *Transportation Science*.

Notteboom, Theo E., & Vernimmen, Bert. 2009. The effect of high fuel costs on liner service configuration in container shipping. *Journal of Transport Geography*, **17**(5), 325–337.

Steenken, Dirk, Voß, Stefan, & Stahlbock, Robert. 2004. Container terminal operation and operations research - A classification and literature review. *Or Spectrum*, **26**(1), 3–49.

Venturini, Giada, Iris, Cagatay, Kontovas, Christos A., & Larsen, Allan. 2017. The multi-port berth allocation problem with speed optimization and emission considerations. *Transportation Research. Part D: Transport and Environment*, **54**, 142–159.