



An exact algorithm for the integrated planning of berth allocation and quay crane assignment

Ilaria Vacca * Matteo Salani[†] Michel Bierlaire *

March 23, 2011

Report TRANSP-OR 110323
Transport and Mobility Laboratory
Ecole Polytechnique Fédérale de Lausanne
`transp-or.epfl.ch`

*Transport and Mobility Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland, { `ilaria.vacca` , `michel.bierlaire` } @epfl.ch

[†]Dalle Molle Institute for Artificial Intelligence (IDSIA), Galleria 2, CH-6928 Manno-Lugano, Switzerland, `matteo.salani@idsia.ch`

Abstract

In this paper we study the simultaneous optimization of berth allocation and quay crane assignment in seaport container terminals. We propose a model based on an exponential number of variables that is solved via column generation. An exact branch-and-price algorithm is implemented to produce optimal integer solutions to the problem. In particular, we present several accelerating techniques for the master and the pricing problem that can be generalized to other branch-and-price schemes. Computational results show that the proposed approach outperforms commercial solvers. Furthermore, the developed algorithm allows for a comparative analysis between the hierarchical and the integrated solution approach that confirms the added value of integration in terms of cost reduction and efficient use of resources. To the best of our knowledge, this is the first exact branch-and-price algorithm for both the berth allocation problem and the berth allocation problem with quay crane assignment.

1 Introduction

Containerized sea-freight transportation has grown dramatically over the last two decades, much faster than other sea transportation modes. Container traffic increased about 9.5% per year between 2000 and 2008, while the average annual rate for cargo traffic was 5.3% (ISL, 2009). The share of containerized trade in the world's total dry cargo increased from 5.1% in 1980 to 25.4% in 2008 (UNCTAD, 2009).

Similarly to air transport and the airline industry, which have greatly benefited from operations research (OR) since the 1950s (Barnhart et al., 2003), maritime transport and seaport logistics represent a more recent OR research field. It has mainly been pushed forward by the competitive environment, forcing operators to optimize their cost to maintain margins. The optimization of container terminal operations has received increasing interest in the scientific literature over the last years and current research directions in container terminal management point towards the integrated planning of operations. This potentially yields to significant improvements in terms of efficiency and productivity for the terminal. From a mathematical point of view, the resulting integrated problems are very complex. Therefore, large-scale optimization techniques need to be designed in order to cope with this complexity and provide exact solutions. In particular, column generation and branch-and-price schemes represent nowadays the most successful tool to solve such complex problems (Lübbecke and Desrosiers, 2005). They have been recently applied to maritime transportation (Grønhaug et al., 2010) and container terminal management (Choo et al., 2010).

In this paper we present an exact algorithm for the integrated planning of berth allocation and quay crane assignment in the context of container terminal management. The problem aims at assigning vessels to berthing positions, performing the scheduling of vessels in each berth and allocating quay cranes (QC) to vessels over a given time horizon, taking into account the quay crane capacity of the terminal. Although different mathematical models and heuristic procedures have been designed for the joint optimization problem (cf. the survey by Bierwirth and Meisel, 2010), no exact approach has been proposed so far.

We propose a model for the integrated optimization problem based on an exponential number of variables. The model exploits the problem

structure and it is solved by column generation. Furthermore, we design and implement an exact branch-and-price algorithm with the purpose of proving good-quality bounds and optimal solutions to the problem. We propose a specific branching scheme and several accelerating techniques both for the pricing and the master problem. Compared to state-of-the-art techniques such as bidirectional dynamic programming and dual stabilization, we reduce the number of states in the pricing problem by defining a new dominance rule and we speed up the dynamic programming (DP) algorithm by introducing an incremental heuristic DP step. These techniques, specifically designed for our problem, proved to be very useful and can be easily generalized to other branch-and-price schemes. Furthermore, our algorithm can be adapted to solve the berth allocation problem only. Therefore, the exact branch-and-price enables us to perform a comparative analysis between the hierarchical and the integrated optimization approach that confirms the added value of integration in terms of cost reduction and efficient use of resources. To the best of our knowledge, this is the first exact branch-and-price algorithm for both the berth allocation problem and the berth allocation problem with quay crane assignment.

The paper is organized as follows. Recent contributions concerning the integrated planning of berth allocation and quay crane assignment are reviewed in section 2. The joint problem is described in section 3, where the mathematical model is presented. The column generation scheme is illustrated in section 4 while section 5 provides details on the implementation of the exact branch-and-price algorithm. Computational results are discussed in section 6, including some insights on the comparison between hierarchical vs integrated solution approaches. Section 7 concludes the paper.

2 Literature review

The need for an efficient management of logistic activities at modern container terminals is well recognized and there exists a rich literature of optimization models and algorithms designed for specific operational problems.

Vis and de Koster (2003) illustrate the main logistic processes in a container terminal reporting about 50 references up to 2001. Steenken et al.

(2004) present an exhaustive overview of optimization methods in container terminal management, reviewing more than 200 references up to 2004; the survey has been recently updated by Stahlbock and Voss (2008). A review of operations research methods in maritime logistics by Christiansen et al. (2004) focuses more generally on ship routing and scheduling.

A promising research track is represented by the integrated optimization of decision problems that are highly interdependent, yet usually solved hierarchically by terminal's planners, as pointed out by Vacca et al. (2007; 2010). In particular, a recent survey by Bierwirth and Meisel (2010) reviews contributions on integrated solution approaches for the berth allocation problem (BAP) and the quay crane assignment problem (QCAP).

The integrated planning of berth allocation and quay crane assignment, introduced by Park and Kim (2003), has been further investigated by Imai et al. (2008) and Meisel and Bierwirth (2009). The resulting models represent a good starting point for tackling such a complex problem; however, they still present some unrealistic assumptions and limits. In fact, the relationship between the number of quay cranes and the handling time is ignored (Imai et al., 2008) or the crane productivity is assumed to be proportional to the number of QCs (Park and Kim, 2003), although it is known that quay cranes interference reduces the marginal productivity.

These unrealistic assumptions are addressed by the model proposed by Giallombardo et al. (2010). The joint problem, called the Tactical Berth Allocation Problem (TBAP), makes use of the concept of quay crane assignment profiles to capture real-world issues. A quay crane profile encodes the number of quay cranes assigned to a vessel along the time steps that the vessel is berthed at the port. The new modeling feature is designed to include some requirements that terminals impose on the quay crane assignment process: quay cranes can be moved from one vessel to another only at the end of a working shift; productivity losses due to crane interference are taken into account in the profile definition as well as vessels' priorities in terms of number of quay cranes and handling time. The authors propose two mixed integer programming (MIP) formulations and a heuristic algorithm based on tabu search and mathematical programming techniques, and provide computational results based on real-world instances.

From an algorithmic point of view, all the mentioned approaches rely

on heuristic methods to provide good and fast solutions. To the best of our knowledge, the only attempt to solve exactly the berth allocation problem (without quay crane assignment) is the one by Buhrkal et al. (2011): the authors present a generalized set-partitioning model for the discrete BAP where all columns are enumerated a-priori. The formulation outperforms existing models (e.g. Imai et al., 2001; Cordeau et al., 2005) and guarantees optimality; however, authors recognize that a branch-and-price algorithm should be implemented in order to solve larger instances.

From the reviewed contributions we remark that there exists no exact algorithm for the integrated planning of berth allocation and quay crane assignment. However, we believe that it is important to characterize optimal solutions and in this paper we intend to exploit the problem structure in order to design an exact algorithm able to cope with the problem complexity.

3 The model

We consider the problem defined by Giallombardo et al. (2010) for the Tactical Berth Allocation Problem.

Given a set of vessels and a set of berths, the joint optimization problem aims at assigning a berthing position and a quay crane assignment profile to every vessel over a given time horizon as well as at scheduling incoming vessels according to their time windows. The objective is to maximize the difference between the revenue associated with the chosen quay crane profiles and the housekeeping costs generated by transshipment flows exchanged by the vessels.

A monetary value is associated with every quay crane profile, which corresponds to the price charged by the terminal to the shipping companies for the provided service.

The key point is that the handling time associated with the vessel directly depends on the number of assigned quay cranes and thus on the assigned profile. Furthermore, the total capacity of the terminal in terms of quay cranes cannot be exceeded.

Time is discretized in time steps of constant length (one hour in our study).

The input data of the Tactical Berth Allocation Problem are:

N	set of vessels;
M	set of berths;
H	set of time steps;
P_i	set of feasible quay crane assignment profiles for vessel $i \in N$;
t_i^p	handling time associated with profile $p \in P_i, i \in N$, expressed as number of time steps;
v_i^p	monetary value associated with the quay crane profile $p \in P_i$;
q_i^{pu}	number of quay cranes used by profile $p \in P_i, i \in N$ at time step $u \in (1, \dots, t_i^p)$;
Q^h	maximum number of quay cranes available at time step $h \in H$;
f_{ij}	number of containers exchanged between vessels $i, j \in N$;
d_{kw}	unit housekeeping cost between yard slots corresponding to berths $k, w \in M$;
$[a_i, b_i]$	[earliest, latest] feasible arrival time step of ship $i \in N$;
$[a^k, b^k]$	[start, end] of availability time step of berth $k \in M$.

Our model is based on the concept of *berth sequence*, a sequentially ordered subset of ships in a berth with an assigned quay crane profile. We define the set Ω^k for every $k \in M$ that represents the set of all sequences of vessels moored at berth k , feasible with respect to time windows constraints and with a unique quay crane profile assigned to each vessel.

In order to model sequences of vessels in berth $k \in M$, the graph $G^k = (V^k, A^k) \forall k \in M$ is defined, where $V^k = N \cup \{o(k), d(k)\}$ and $o(k)$ and $d(k)$ are additional vertices representing berth k . The set of arcs $A^k \subseteq V^k \times V^k$ represents feasible precedence relationships between vessels: an arc (i, j) exists if vessel i can precede vessel j according to their time windows.

The decision variables of our optimization problem are:

- $x_{ij}^k \in \{0, 1\} \forall k \in M, \forall (i, j) \in A^k$, equal to 1 if ship j is scheduled after ship i at berth k , and 0 otherwise;
- $y_i^k \in \{0, 1\} \forall k \in M, \forall i \in N$, equal to 1 if ship i is assigned to berth k , and 0 otherwise;
- $\lambda_i^p \in \{0, 1\} \forall p \in P_i, \forall i \in N$, equal to 1 if ship i is served by the profile p , and 0 otherwise;
- $s_{r_k} \geq 0 \forall r_k \in \Omega^k$, associated with the selection of sequence r_k . At optimality, s_{r_k} assumes binary values: it equals 1 if sequence r_k is chosen in the optimal solution and 0 otherwise.

Every berth sequence $r_k \in \Omega^k$ is described by the following coefficients:

- x_{ijr_k} binary coefficient equal to 1 if vessel j follows vessel i in sequence r_k ;
- y_{ir_k} binary coefficient equal to 1 if vessel i is moored at berth k in sequence r_k ;
- $\lambda_{ir_k}^p$ binary coefficient equal to 1 if profile p is assigned to vessel i in sequence r_k ;
- $q_{r_k}^h$ coefficient that counts the number of quay cranes used by sequence r_k at time step h ;
- v_{r_k} monetary value associated with sequence r_k , defined as

$$v_{r_k} = \sum_{i \in N} \sum_{p \in P_i} \lambda_{ir_k}^p v_i^p.$$

The Tactical Berth Allocation Problem is formulated as follows:

$$\max \sum_{k \in M} \sum_{r_k \in \Omega^k} v_{r_k} s_{r_k} - \sum_{i,j \in N} \sum_{k,w \in M} f_{ij} d_{kw} y_i^k y_j^w \quad (1)$$

$$\sum_{k \in M} \sum_{r_k \in \Omega^k} y_{ir_k} s_{r_k} = 1 \quad \forall i \in N, \quad (2)$$

$$\sum_{k \in M} \sum_{r_k \in \Omega^k} q_{r_k}^h s_{r_k} \leq Q^h \quad \forall h \in H, \quad (3)$$

$$\sum_{r_k \in \Omega^k} s_{r_k} \leq 1 \quad \forall k \in M, \quad (4)$$

$$\sum_{r_k \in \Omega^k} x_{ijr_k} s_{r_k} = x_{ij}^k \quad \forall i, j \in N, \forall k \in M, \quad (5)$$

$$\sum_{r_k \in \Omega^k} y_{ir_k} s_{r_k} = y_i^k \quad \forall i \in N, \forall k \in M, \quad (6)$$

$$\sum_{k \in M} \sum_{r_k \in \Omega^k} \lambda_{ir_k}^p s_{r_k} = \lambda_i^p \quad \forall p \in P_i, \forall i \in N, \quad (7)$$

$$s_{r_k} \geq 0 \quad \forall r_k \in \Omega^k, \forall k \in M, \quad (8)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A^k, \forall k \in M, \quad (9)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in N, \forall k \in M, \quad (10)$$

$$\lambda_i^p \in \{0, 1\} \quad \forall p \in P_i, \forall i \in N. \quad (11)$$

The objective function (1) maximizes the difference between the total monetary value associated with the selected sequences, i.e., the total value of selected profiles, and the total housekeeping cost generated by the berth allocation plan. Constraints (2) ensure that every ship is assigned to exactly one sequence, and thus to one berth, while constraints (3) ensure that the quay crane capacity is not violated. Constraints (4) select at most one sequence for each berth. Constraints (5)-(8) link decision variables x_{ij}^k ,

y_i^k and λ_i^p to s_{r_k} . Finally, the integrality of the solution is ensured by constraints (9)-(11).

The proposed model relies on an exponential number of variables s_{r_k} and it is therefore solved by the means of column generation. To this purpose, the quadratic objective function is linearized by adding a new decision variable $z_{ij}^{kw} = y_i^k y_j^w$, as suggested by Giallombardo et al. (2010).

Furthermore, we remark that model (1)–(11) corresponds to the Dantzig-Wolfe reformulation of the TBAP model by Giallombardo et al. (2010). Indeed, constraints (5)-(8) represent the so-called *coupling constraints* that make the link between variables of the compact formulation (x_{ij}^k , y_i^k and λ_i^p) and variables of the extensive formulation (s_{r_k}).

4 Column generation

The linear relaxation of formulation (1)–(11), linearized by the means of z_{ij}^{kw} variables, is solved via column generation. In this section we define the master problem and the pricing subproblem and we illustrate the column generation scheme.

4.1 Master problem

If we relax the integrality requirements (9)-(11), constraints (5)-(7) become redundant and can be removed from the formulation. We obtain the following master problem:

$$\max \sum_{k \in M} \sum_{r_k \in \Omega^k} v_{r_k} s_{r_k} - \sum_{i, j \in N} \sum_{k, w \in M} f_{ij} d_{kw} z_{ij}^{kw} \quad (12)$$

$$\sum_{k \in M} \sum_{r_k \in \Omega^k} y_{ir_k} s_{r_k} = 1 \quad \forall i \in N, \quad (13)$$

$$\sum_{k \in M} \sum_{r_k \in \Omega^k} q_{r_k}^h s_{r_k} \leq Q^h \quad \forall h \in H, \quad (14)$$

$$\sum_{r_k \in \Omega^k} s_{r_k} \leq 1 \quad \forall k \in M, \quad (15)$$

$$\sum_{k \in M} \sum_{w \in M} z_{ij}^{kw} = g_{ij} \quad \forall i, j \in N, \quad (16)$$

$$\sum_{r_k \in \Omega^k} y_{ir_k} s_{r_k} - z_{ij}^{kw} \geq 0 \quad \forall i, j \in N, \forall k, w \in M \quad (17)$$

$$\sum_{r_w \in \Omega^w} y_{jr_w} s_{r_w} - z_{ij}^{kw} \geq 0 \quad \forall i, j \in N, \forall k, w \in M \quad (18)$$

$$z_{ij}^{kw} \geq 0 \quad \forall i, j \in N, \forall k, w \in M \quad (19)$$

$$s_{r_k} \geq 0 \quad \forall r_k \in \Omega^k, \forall k \in M. \quad (20)$$

where constraints (16)–(19) are due to the linearization of objective function (1).

The resulting linear program involves an exponential number of variables (columns). Therefore, the column generation scheme starts solving a *restricted master problem*, defined on a subset of columns and, at each iteration, it generates new profitable columns to be added to the formulation, if any.

4.2 Pricing subproblem

Let π , μ , ξ , θ and η be the dual vectors associated with constraints (13), (14), (15), (17) and (18), respectively. Given an optimal solution of the restricted master problem, the reduced cost of sequence $r \in \Omega^k$ is given by:

$$\tilde{v}_{r_k} = v_{r_k} - \sum_{i \in N} \pi_i y_{ir_k} - \sum_{h \in H} \mu^h q_{r_k}^h - \xi_k - \sum_{i, j \in N} \sum_{w \in M} \theta_{ij}^{kw} y_{ir_k} - \sum_{i, j \in N} \sum_{w \in M} \eta_{ij}^{kw} y_{jr_w}.$$

The pricing subproblem identifies, for every berth $k \in M$, the column r_k^* with the maximum reduced cost. Additional decision variables are:

- $T_i \geq 0$, representing the arrival time step of ship $i \in N$;
- $T_o \geq 0$, representing the time step when operations start in the berth;
- $T_d \geq 0$, representing the time step when operations end in the berth.

The pricing subproblem is formulated as follows:

$$\max \sum_{i \in N} (v_i^p \lambda_i^p - \pi_i y_i) - \sum_{i \in N} \sum_{p \in P_i} \sum_{h=1 \dots t_i^p} \mu^{T_i+h-1} q_i^{ph} \lambda_i^p - \xi_k - \sum_{i,j \in N} \sum_{w \in M} (\theta_{ij}^{kw} y_i + \eta_{ij}^{kw} y_j) \quad (21)$$

$$\sum_{j \in N \cup \{d\}} x_{o,j} = 1, \quad (22)$$

$$\sum_{i \in N \cup \{o\}} x_{i,d} = 1, \quad (23)$$

$$\sum_{j \in N \cup \{d\}} x_{ij} - \sum_{j \in N \cup \{o\}} x_{ji} = 0 \quad \forall i \in N, \quad (24)$$

$$\sum_{j \in N \cup \{d\}} x_{ij} = y_i \quad \forall i \in N, \quad (25)$$

$$T_i + \sum_{p \in P_i} t_i^p \lambda_i^p - T_j \leq (1 - x_{ij}) M1 \quad \forall i \in N, \forall j \in N \cup \{d(k)\}, \quad (26)$$

$$T_o - T_j \leq (1 - x_{o,j}) M2 \quad \forall j \in N, \quad (27)$$

$$a_i y_i \leq T_i \quad \forall i \in N, \quad (28)$$

$$T_i \leq b_i y_i \quad \forall i \in N, \quad (29)$$

$$a^k \leq T_o, \quad (30)$$

$$T_d \leq b^k, \quad (31)$$

$$\sum_{p \in P_i} \lambda_i^p = y_i \quad \forall i \in N, \quad (32)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A^k, \quad (33)$$

$$y_i \in \{0, 1\} \quad \forall i \in N, \quad (34)$$

$$\lambda_i^p \in \{0, 1\} \quad \forall p \in P_i, \forall i \in N, \quad (35)$$

$$T_i \geq 0 \quad \forall i \in N \cup \{o, d\}, \quad (36)$$

where M1 and M2 are large positive constants.

The objective function (21) maximizes the reduced cost of a column in berth k . Constraints (22)–(24) ensures flow conservation, while variables y_i are linked to variables x_{ij} by constraints (25). Precedence constraints (26)–(27) and time windows constraints (28)–(31) ensure the correct scheduling of vessels over time while the profile assignment is controlled by the set of constraints (32). Finally, constraints (33)–(36) define the domain of variables.

We remark that index k has disappeared from decision variables x and y with respect to the notation introduced in section 3, since the pricing subproblem is solved for a fixed berth k .

At each iteration of column generation, we solve $m = |M|$ subproblems, one for every berth $k \in M$. If $\tilde{v}_{r_k^*} > 0$ for some k , we add column $s_{r_k^*}$ to the restricted master problem and we iterate the process; otherwise, the current solution of the master problem is proven to be optimal and we stop.

4.2.1 Dynamic programming

The pricing subproblem (21)–(36) can be cast to a Resource Constrained Elementary Shortest Path Problem (RCESPP), where the resource is represented by time, and it is solved by the means of dynamic programming. The underlying network $G(V, A)$ has one vertex for every vessel $i \in N$, for every profile $p \in P_i$ and for every time step $h \in H$. Transit time on arcs equals the handling time of profile $p \in P_i$ assigned to vessel i . Vertex (i, h, p) represents vessel i berthed at time step h and operated by quay crane profile p . The graph has two additional vertices o, d associated with the specific berth $k \in M$ for which the pricing is solved, representing the origin and the destination of the path.

The RCESPP aims at finding a maximum-cost elementary path from o to d that satisfies the constraints on resources: the objective function of the RCESPP associated with the pricing subproblem corresponds to (21), while the resource constraint requires not to exceed the given time horizon.

The dynamic programming (DP) algorithm iteratively extends states. A *state* for vertex (i, h, p) represents a path from o to (i, h, p) ; many states are associated with the same vertex (i, h, p) , representing different paths. Each state is encoded by a *label* of the form (S, τ, C, i, h, p) , that is a path

from o to (i, h, p) with time consumption τ and cost C ; furthermore, to ensure elementarity, set S keeps tracks of vessels visited along the path (Beasley and Christofides, 1989). The optimal solution is given by the maximum cost state associated with the destination vertex d .

At vertex o , time consumption τ is initialized at 0 and $S = \{o\}$; cost C is initialized to $-\xi_k$, where k is the berth for which the pricing problem is being solved. When extending state $(S, \tau, C, i, h_i, p_i)$ to another feasible state $(S', \tau', C', j, h_j, p_j)$, the label is updated according to the formula:

$$S' = S + \{j\} \quad (37)$$

$$\tau' = \tau + t_j^{p_j} \quad (38)$$

$$C' = C + v_{p_j} - \pi_j - \sum_{h=h_j}^{h_j+t_j^{p_j}} \mu^h q_j^{p_j(h-h_j)} - \sum_{n \in N} \sum_{w \in M} (\theta_{jn}^{kw} + \eta_{nj}^{wk}) \quad (39)$$

The extension is feasible if $j \notin S$ (elementarity), $\tau' < |H|$ (total duration) and h_j satisfies time windows $[a_j, b_j]$.

The efficiency of dynamic programming strongly depends on the effectiveness of *dominance* rules that are used to fathom feasible, yet non-optimal states. In particular, dominated states are not extended further. According to Beasley and Christofides (1989), state $(S', \tau', C', j, h_j, p_j)$ dominates $(S'', \tau'', C'', j, h_j, p_j)$ if:

$$S' \leq S'' \quad (40)$$

$$\tau' \leq \tau'' \quad (41)$$

$$C' \geq C'' \quad (42)$$

and at least one of these inequalities is strictly satisfied.

5 Branch-and-price algorithm

In order to obtain integer solutions, we implement a branch-and-price algorithm where column generation is applied at every node of the search tree. The search tree is explored according to a best-first strategy with respect to the upper bound associated with the node. The algorithm makes use of a column pool that keeps track of all columns generated in different nodes of the search tree.

In the remainder of this section we illustrate the branching rules as well as accelerating techniques both for the pricing and the master problem.

5.1 Branching scheme

In the search tree, branching is required when the master problem is solved at optimality and the corresponding solution in terms of original formulation's variables is not integer. We implement a branching scheme consisting of four hierarchical levels.

1. If the total number of berths $\tilde{K} = \sum_{k \in M} \sum_{r_k \in \Omega^k} s_{r_k}$ is fractional, then branching requires an additional constraint to be added in the master problem:

- $\sum_{k \in M} \sum_{r_k \in \Omega^k} s_{r_k} \leq \lfloor \tilde{K} \rfloor$ on the first child node;
- $\sum_{k \in M} \sum_{r_k \in \Omega^k} s_{r_k} \geq \lceil \tilde{K} \rceil$ on the second child node.

This branching requires the dual value associated with the additional constraint, denoted by π_0 , to be collected and accounted in the pricing subproblem. We remark that π_0 is a constant, regardless of the berth. In particular, the additional constraint in the master problem modifies the objective function of the pricing subproblem as follows:

$$\max v_{r_k} - \sum_{i \in N} \pi_i y_{ir_k} - \sum_{h \in H} \mu^h q_{r_k}^h - \xi_k - \sum_{i,j \in N} \sum_{w \in M} (\theta_{ij}^{kw} y_{ir_k} + \eta_{ij}^{kw} y_{jr_w}) - \pi_0.$$

2. If some vessel $i \in N$ is assigned fractionally to some berth $k \in M$, i.e., quantity $\tilde{Y}_i^k = \sum_{r_k \in \Omega^k} y_i^{r_k} s_{r_k}$ is fractional, then the branching requires an additional constraint to be added to the master problem for the given vessel i and berth k :

- $\sum_{r_k \in \Omega^k} y_i^{r_k} s_{r_k} = 0$ on the first child node;
- $\sum_{r_k \in \Omega^k} y_i^{r_k} s_{r_k} = 1$ on the second child node.

This branching requires the dual value associated with the additional constraint, denoted by φ_i^k , to be taken into account in the pricing subproblem. We remark that φ_i^k is collected in the pricing for berth k if vessel i is visited by the sequence. In particular, the additional

constraints in the master problem modify the objective function of the pricing subproblem as follows:

$$\max v_{r_k} - \sum_{i \in N} \pi_i y_{i r_k} - \sum_{h \in H} \mu^h q_{r_k}^h - \xi_k - \sum_{i, j \in N} \sum_{w \in M} (\theta_{ij}^{kw} y_{i r_k} + \eta_{ij}^{kw} y_{j r_w}) - \pi_0 - \sum_{i \in N} \varphi_i^k y_{i r_k}.$$

3. If some profile $p \in P_i$ is assigned fractionally to some vessel $i \in N$, i.e., quantity $\tilde{\lambda}_i^p = \sum_{k \in M} \sum_{r_k \in \Omega^k} \lambda_i^{p r_k} s_{r_k}$ is fractional, then branching is handled directly in the pricing subproblem by modifying the set P_i of feasible profiles for vessel i . On the first node child, we enforce profile p to be assigned to vessel i by removing all other feasible profiles from set P_i ; this branching corresponds to enforce $\lambda_i^p = 1$ in the original formulation. On the second child node, we prevent profile p to be used by removing it by set P_i ; this branching corresponds to enforce $\lambda_i^p = 0$ in the original formulation. We remark that neither the master nor the pricing formulation is modified by this branching in terms of objective function and additional constraints. However, infeasible columns must be removed from the master problem, according to the branching decision associated with the analyzed node.
4. If none of the above conditions holds, then there exist some vessel $i \in N$ such that the quantity $\tilde{T}_i^h = \sum_{k \in M} \sum_{r_k \in \Omega^k} T_i^{h r_k} s_{r_k}$ is fractional for some $h^* \in H$, where $T_i^{h r_k}$ is a binary coefficient equal to 1 if vessel i arrives at time step h in sequence r_k . In this case, branching is handled in the pricing subproblem, by modifying the time windows $[a_i, b_i]$ associated with vessel i , as similarly proposed by Gélinas et al. (1995) for the Vehicle Routing Problem. We denote t_i^* the arrival time associated with time step h^* . On the first child node, we enforce the vessel to arrive before time step h^* : the new time windows for vessel i are therefore $[a_i, t_i^* - \epsilon]$ and this corresponds to enforce $\gamma_i^h = 0 \forall h \geq h^*$ in the original formulation. On the second child node, we enforce the vessel to arrive at or after time step h^* : the new time windows for vessel i are therefore $[t_i^*, b_i]$ and this corresponds to enforce $\gamma_i^h = 0 \forall h < h^*$ in the original formulation. We remark that neither the master nor the pricing formulation is modified by this branching in terms of objective function and additional constraints. However, infeasible columns must be removed from the master problem, according to the branching decision associated with the analyzed node.

The order of branching is determined by the increasing complexity of the branching rules due to additional constraints in the master problem or additional complexity of the pricing problem.

5.2 Accelerating techniques

5.2.1 Solving exact dynamic programming

We implement state-of-the-art techniques for solving the RCESPP such as bounded bidirectional dynamic programming and decremental state space relaxation.

Bounded bidirectional DP (Righini and Salani, 2006) consists of two steps: firstly, states are extended in forward and backward direction until half of the so-called *critical resource* (time, in our case) is consumed; secondly, forward and backward paths are joined to produce feasible sequences. Bounding is used to discard non-dominated non-optimal states.

The basic idea of decremental state space relaxation (Righini and Salani, 2008) is to start checking elementarity only on a subset \bar{S} of S . If the final solution is non-elementary, one or more vertices violating the constraint are added to \bar{S} and DP is executed again.

The implemented search policy takes into account time windows (Liberatore et al., 2011). At every iteration of dynamic programming, states are explored according to the vertices (vessels) they are associated with. We decide to sort vessels according to the starting time a_i of their time windows; this search strategy proves to be important for the effectiveness of the algorithm in our tests.

Furthermore, we design an additional technique for accelerating the exact pricing, that is specifically designed for our pricing subproblem.

Domination of (h,p) pairs Unlike RCESPP subproblems arising in vehicle routing, where customers are visited one right after the other, in our problem it may be convenient to wait some time between the departure of a vessel and the arrival of the next one. This is due to the quay crane capacity constraint, that control the interactions between berths at the master problem level; in particular, these interactions are captured by dual vectors θ and η . More specifically, when extending a label to the next vessel j , we have as many new states as the number of feasible arrival time steps

h_j ; furthermore, we may have more than one profile p_j associated with a single time step h_j and viceversa. In order to reduce the number of states, preprocessing is performed at the beginning of the DP algorithm: we populate a list of non-dominated (h_j, p_j) pairs for every vessel j and we refer to this list when extending a label to vessel j . We remark that in the special case of $\theta = 0$ and $\eta = 0$, the list has at most one pair (h_j, p_j) for every profile p_j .

5.2.2 Heuristic pricing

The pricing subproblem is firstly solved heuristically. An exact solution is computed only if needed.

The heuristic dynamic programming algorithm is based on a relaxed dominance rule that allows to eliminate much more states during the comparison of labels (Dell'Amico et al., 2006). The final solution is an elementary shortest path that satisfies resource constraints; however, optimality is no longer guaranteed. When using relaxed dominance, we say that state $(S', \tau', C', j, h_j, p_j)$ dominates $(S'', \tau'', C'', j, h_j, p_j)$ if:

$$\tau' \leq \tau'' \quad (43)$$

$$C' \geq C'' \quad (44)$$

and at least one of these inequalities is strictly satisfied. In other words, we do not compare anymore the set of vertices S', S'' visited by the partial paths. As the dominance is weaker, the number of eliminated labels is greater. This results in a reduced computational effort to solve the pricing. In particular, we remove from the dominance criteria the set S of visited nodes as it encodes the combinatorial nature of the problem. After its removal the solution can be found in pseudo-polynomial time.

Furthermore, the following accelerating techniques are implemented with the main purpose of avoiding the call to exact pricing as much as possible.

Multiple pricing strategy At every iteration of column generation, we firstly solve a pricing subproblem for every berth $k \in M$ using the heuristic dynamic programming algorithm; exact DP is called only if heuristic pricing cannot provide a negative reduced cost column. As soon as we

find a negative reduced cost column for some berth k^* , the pricing terminates. At this point, columns generated for berth k^* are evaluated for all berths $k \neq k^*, k \in M$: if a column is feasible for another berth, say \bar{k} , and its reduced cost re-computed for berth \bar{k} is strictly positive, then the column is duplicated and added to the master problem also for berth \bar{k} . This strategy provides a fast column generation, avoiding the time-consuming pricing subproblem for berth \bar{k} .

Incremental heuristic dynamic programming The basic idea is to incrementally strengthen the relaxed dominance rule introduced for the heuristic pricing, in order to increase the probability of finding a negative reduced cost column and therefore avoid calling exact DP. We define the set of *critical vertices* $\tilde{N} \subset N$ for which exact dominance is required, similarly to decremental state space relaxation (Righini and Salani, 2008); the set \tilde{N} is initialized with the empty set, and it is iteratively incremented until a given percentage δ of vertices is reached. At each iteration, $\beta|N|$ critical vertices are chosen from the set $N \setminus \tilde{N}$. Vertices with the greatest associated non-negative dual price are chosen. The dominance rule is the one described in section 4.2, except for the definition of set S : a vertex j belongs to S if it is visited by the partial path and if $j \in \tilde{N}$. The first iteration, when $\tilde{N} = \{\emptyset\}$, corresponds to the heuristic DP algorithm outlined at the beginning of this subsection; the special case of $\delta = 1$ corresponds to exact dynamic programming. In our tests, we fix $\beta = 0.2$ and $\delta = 0.4$.

5.2.3 Dual stabilization

Column generation is known to suffer slow convergence (*tailing-off effect*) mainly due to stability problem. Degeneracy of the master problem implies an infinite number of dual optimal solutions: the simplex method typically provides an extreme dual optimal vector, whereas interior dual vectors could be more suitable for generating good paths in the pricing subproblem. Stabilization methods try to overcome this issue by providing a better approximation of optimal dual values (du Merle et al., 1999; Rousseau et al., 2007).

Our stabilized version of column generation is inspired by Addis et al. (2009). The basic idea is the following: a dual optimal solution π to the restricted master problem can be either feasible, and thus optimal, or in-

feasible for the dual of the complete master problem. We are mainly interested in pricing out with a dual vector close to the optimal dual, thus close to feasibility.

We define the *stability center* $\tilde{\pi}$ that represents our current best guess for the optimal dual. At each iteration of column generation, we modify the dual vector provided by the restricted master problem and we obtain a new vector $\tilde{\pi}$ that we use in the pricing problem. The update formula is clear and simple:

$$\tilde{\pi} = \alpha\pi + (1 - \alpha)\tilde{\pi}, \quad (45)$$

where α is a parameter between 0 and 1. At every iteration of column generation the value of α is initialized to $\alpha_0 = 0.5$ and it is increased by a step of 0.1 until positive reduced cost columns are found. The process is repeated until $\alpha = 1$ and no positive reduced cost columns can be found.

5.2.4 Primal heuristic

Integer feasible solution are rarely produced in column generation, as the optimal solutions of restricted master problems are typically fractional. Therefore we implement a primal heuristic in order to identify feasible integer solutions during the search process: the main purpose is to improve the primal bound, and thus increase the pruning in the search tree.

The heuristic algorithm takes as input a fractional optimal solution to a restricted master problem and identifies the variable $s_{r_k}^*$ with the highest fractional value strictly lower than 1; variable $s_{r_k}^*$ is set equal to 1 and the linear program is solved again. The procedure is repeated until either a integer solution is found or the linear problem becomes infeasible.

Although very simple, the primal heuristic proves to be helpful in finding integer solutions especially for larger instances.

5.2.5 Initialization

The master problem is initialized with a set of artificial variables that satisfy constraints (13), (17) and (18).

A second initialization makes use of the solution provided by the heuristic algorithm by Giallombardo et al. (2010): columns associated to the initial solution are added to the master problem at the root node.

6 Computational results

The branch-and-price algorithm for the TBAP is implemented in C++ and compiled with gcc 4.1.2. All restricted master problems are solved using ILOG CPLEX version 12. Computational experience is run under a linux operating system on a 2Ghz Intel processor equipped with 2GB of RAM.

6.1 Instances

Computational experiments are performed on instances derived by the test set introduced by Giallombardo et al. (2010). We consider instances with 10 up to 20 vessels and 3 up to 5 berths over a time horizon of one week. The name of the instance indicates the traffic volume, high (H) or low (L), and the number of feasible quay crane assignment profiles for each vessel (10, 20 or 30).

In addition to the existing instances, an intermediate class of instances is defined. The new set considers 15 vessels and 3 berths over a time horizon of one week; these instances are obtained by considering the first 15 vessels and the first 3 berths of the class 20 x 3. Furthermore, we consider an additional set of instances for class 20 x 5 where the time horizon is shortened from seven to four working days; this new class is denoted by the suffix 4d in the instance name.

6.2 Branch-and-price results

We compare our branch-and-price algorithm to the mixed integer linear programming (MILP) formulation by Giallombardo et al. (2010) solved by CPLEX. The results presented by Giallombardo et al. (2010) have been re-performed on our machines with CPLEX version 12.

Tables 1 and 2 report results for instances with 10 vessels and 3 berths over a time horizon of one week. Table 1 provides a comparison between the upper bound of the linear relaxation of the original MILP formulation (z_{LP}) and the upper bound obtained via Dantzig-Wolfe reformulation (z_{DW}), i.e., the optimal value of the master problem at the end of the root node. Computational times are not reported, as they are negligible for both cases. Column ' $\% z_{DW}$ ' reports the percentage of the bound improvement, that is always less than 0.5%, thus not very significant. However,

10 x 3 Instance	z_{LP}	z_{DW}	% z_{DW}
10_3_H1_p10	800614	797594	0.38%
10_3_H1_p20	800890	797870	0.38%
10_3_H1_p30	800924	798136	0.35%
10_3_H2_p10	740947	738540	0.32%
10_3_H2_p20	741487	739190	0.31%
10_3_H2_p30	741523	739417	0.28%
10_3_L1_p10	519319	518334	0.19%
10_3_L1_p20	519354	518750	0.12%
10_3_L1_p30	519389	518785	0.12%
10_3_L2_p10	568152	566976	0.21%
10_3_L2_p20	568188	567012	0.21%
10_3_L2_p30	568224	567146	0.19%

Table 1: *Linear relaxation results for 10 ships and 3 berths over 1 week.*

the DW formulation proves to be much stronger than MILP when embedded into a branch-and-bound framework. Indeed, we notice that, after one hour of computation, the CPLEX bound is unchanged, despite of the depth of the search tree and the several branching decisions that have been made. On the contrary, it is often sufficient to perform a few branching decisions to see an improvement in the bound provided by the master problem.

The superiority of our approach is clearly shown in Table 2, that compares our branch-and-price algorithm to the general-purpose MIP solver on instances with 10 vessels; the root node initialization relies either on the artificial variables only (B&P) or on the solution provided by the heuristic by Giallombardo et al. (2010) (B&P + INIT). Column `opt_sol` reports the value of the optimal solution while column `t(s)` reports the computational time of branch-and-price expressed in seconds; the time required by the heuristic algorithm to provide the initial solution is denoted by `t(init)`, while the total computational time (initialization step + branch-and-price) is denoted by `t(tot)`. The best solution found by CPLEX in one hour of computation is denoted by `best_sol`; the gap with respect to the optimal

10 x 3 Instance	CPLEX (1h)		B&P		B&P + INIT			
	best_sol	GAP	opt_sol	t(s)	opt_sol	t(s)	t(init)	t(tot)
H1_p10	x	∞	790735	21	790735	16	7	23
H1_p20	x	∞	791011	25	791011	15	21	36
H1_p30	780722	1.30%	791045	10	791045	25	39	64
H2_p10	712669	2.81%	733276	2	733276	7	8	15
H2_p20	x	∞	735646	7	735646	9	20	29
H2_p30	723818	1.61%	735682	9	735682	8	33	41
L1_p10	515902	0.00%	515902	7	515902	11	7	18
L1_p20	515991	0.40%	518049	5	518049	11	18	29
L1_p30	513731	0.84%	518084	27	518084	125	37	162
L2_p10	564831	0.00%	564831	9	564831	8	8	16
L2_p20	561504	0.60%	564867	7	564867	19	18	37
L2_p30	559389	0.98%	564903	8	564903	20	36	56

Table 2: *Branch-and-price results for 10 ships and 3 berths over 1 week.*

solution value is also reported.

The branch-and-price algorithm clearly outperforms CPLEX: it always provides the optimal solution in a few seconds, whereas the MIP solver often produces feasible solutions within a gap of 3%. In three cases, the MILP formulation cannot find a feasible solution in one hour. Furthermore, we remark that CPLEX fails to prove optimality for instances L1_p10 and L2_p10 because of the poor linear relaxation bound, that cannot be improved during the search in the branch-and-bound tree. Remarkably, for this class of instances, our branch-and-price shows computational times comparable (or even smaller) than the heuristic algorithm, while ensuring optimality of the solutions. As a result, the heuristic initialization slows down our branch-and-price algorithm. Still, the superiority with respect to CPLEX is evident.

Computational results for instances with 15 vessels and 3 berths over a time horizon of one week are reported in Tables 3 and 4. The improvement of the linear relaxation bound is comparable to the one obtained for class 10 x 3 (about 0.5%), and the computational time, although not negligible,

15 x 3 Instance	z_{LP}	z_{DW}	% z_{DW}	t(s)
15_3_H1_p10	1189962	1183344	0.56%	11
15_3_H2_p10	1292357	1285075	0.56%	11
15_3_L1_p10	1110159	1105395	0.43%	7
15_3_L2_p10	899876	896483	0.38%	320

Table 3: *Linear relaxation results for 15 ships and 3 berths over 1 week.*

15 x 3 Instance	CPLEX (1h)		B&P		B&P + INIT			
	best_sol	GAP	opt_sol	t(s)	opt_sol	t(s)	t(init)	t(tot)
H1_p10	x	∞	1170783	3507	1170783	1448	34	1482
H2_p10	1250124	3.27%	1272236	3787	1272236	1673	32	1704
L1_p10	x	∞	1098411	1203	1098411	898	116	1014
L2_p10	x	∞	890211	8975	890211	3555	28	3583

Table 4: *Branch-and-price results for 15 ships and 3 berths over 1 week.*

is still reasonable (except for instance L2_p10). The branch-and-price algorithm always finds the optimal solution. CPLEX is only able to provide a feasible solution for instance H2_p10 within 3 hours of computational time. Although the branch-and-price algorithm requires a significant effort in terms of computational time (about 2 orders of magnitude higher than for class 10 x 3), the heuristic initialization is very effective in speeding up the whole procedure: computational time is more than halved for instances H1, H2 and L2 and reduced by a third for instance L1. All instances are therefore solved within one hour.

Tables 5 and 6 report results for instances with 20 vessels and 5 berths over a time horizon of 4 days. The linear relaxation bound is improved on average by 2%; remarkably, the MILP linear relaxation bound is unchanged, despite of the time horizon reduction and modified time windows. This gives an insight of how much “fractional” the MILP linear relaxation solution is. The root node is closed in about 5 minutes on average. In Table 6 we report the best solutions found after three hours of computation: CPLEX is not able to provide any feasible solution, whereas the branch-and-price algorithm always provides solution with a gap between 3% and 5%. Furthermore, the heuristic initialization produces a gap reduction for all the instances.

Finally, tables 7 and 8 report the results for instances with 20 vessels

20 x 5 Instance	z_{LP}	z_{DW}	% z_{DW}	t(s)
20_5_H1_p10_4d	1383614	1356460	1.96%	315
20_5_H2_p10_4d	1474082	1444042	2.04%	136
20_5_L1_p10_4d	1298356	1274821	1.81%	483
20_5_L2_p10_4d	1103212	1084936	1.66%	373

Table 5: *Linear relaxation results for 20 ships and 5 berths over 4days.*

20 x 5 Instance	CPLEX (3h)		B&P (3h)		B&P + INIT (3h)	
	best_sol	GAP	best_sol	GAP	best_sol	GAP
H1_p10_4d	x	∞	1293184	4.66%	1305216	3.78%
H2_p10_4d	x	∞	1379208	4.49%	1381241	4.35%
L1_p10_4d	x	∞	1224458	3.95%	1231385	3.41%
L2_p10_4d	x	∞	1045778	3.61%	1050171	3.20%

Table 6: *Branch-and-price results for 20 ships and 5 berths over 4days.*

20 x 5 Instance	z_{LP}	z_{DW}	% z_{DW}	t(s)
20_5_H1_p10	1383614	1369818	1.00%	721
20_5_H2_p10	1474082	1459341	1.00%	504
20_5_L1_p10	1298356	1287080	0.87%	520
20_5_L2_p10	1103212	1094480	0.79%	640

Table 7: *Linear relaxation results for 20 ships and 5 berths over 1 week.*

20 x 5 Instance	CPLEX (3h)		B&P (3h)		B&P + INIT (3h)	
	best_sol	GAP	best_sol	GAP	best_sol	GAP
H1_p10	x	∞	x	∞	1337077	2.39%
H2_p10	x	∞	x	∞	1429249	2.06%
L1_p10	1221191	5.12%	1256529	2.37%	1258150	2.25%
L2_p10	x	∞	1059231	3.22%	1070543	2.19%

Table 8: *Branch-and-price results for 20 ships and 5 berths over 1 week.*

10 x 3 Instance	Basic B&P		Improved B&P		speed up(%)
	t(s)	% pricing	t(s)	% pricing	
H1_p10	114	97%	21	10%	82%
H1_p20	995	97%	25	12%	97%
H1_p30	557	99%	10	18%	98%
H2_p10	12	82%	2	10%	83%
H2_p20	29	90%	7	11%	76%
H2_p30	25	92%	9	13%	65%
L1_p10	4054	99%	7	42%	100%
L1_p20	761	99%	5	62%	99%
L1_p30	470	99%	27	93%	94%
L2_p10	4697	99%	9	54%	100%
L2_p20	1573	99%	7	62%	100%
L2_p30	2680	99%	8	63%	100%

Table 9: *Reduction of computational time obtained with the accelerating techniques.*

and 5 berths over a time horizon of one week. In Table 7 we can observe that the improvement in terms of linear relaxation bound has slightly decreased (from 2% to about 1% on average) compared to the 4 days case. Furthermore, the computational effort required by column generation has increased, as it takes about 10 minutes on average to close the root node. On the contrary, the MILP linear relaxation is solved in fractions of a second.

Table 8 compares the best solutions found by CPLEX and our branch-and-price algorithm after three hours of computation. We notice that the increased size of the problem affects the efficiency of the algorithm, since no instance is solved at optimality. However, branch-and-price still performs better than CPLEX: it always provides feasible solutions with a gap between 2% and 3%, whereas CPLEX finds a feasible solution only for instance L1. Furthermore, the importance of a good initialization strategy is emphasized by the results.

Summing up, computational experiments show that our specialized

branch-and-price algorithm outperforms the general-purpose solver in terms of quality of solutions and computational time. The results confirm that designing sophisticated algorithms and exploiting the problem structure is crucial when tackling large-scale optimization problems as the TBAP.

Most of the implemented accelerating techniques concern the pricing problem. This is motivated by the fact that preliminary results produced with a “basic” implementation of the branch-and-price algorithm (that included only heuristic pricing as accelerating technique) point out that about 99% of the computational time was spent in the pricing. The design of sophisticated and specialized techniques for the pricing problem is extremely successful: we reduced the computational time by 90% on average, as shown in Table 9. For the basic and the specialized implementation of the branch-and-price algorithm we report the computational time ($t(s)$) and the percentage of time spent in solving the pricing problem ($\%pricing$). The results confirm the large speed up produced by the designed accelerating techniques on the solution process.

6.3 Hierarchical vs integrated planning

The exact branch-and-price algorithm enables us to perform a comparative analysis between hierarchical and integrated optimization approaches for the Tactical Berth Allocation Problem. In hierarchical planning, berth allocation is solved first, according to an estimated handling time for the vessels; in a second stage, the quay crane assignment is performed on the resulting berth allocation plan. On the contrary, in the integrated planning approach, the two problems are solved simultaneously.

Remarkably, our algorithm can be adapted to solve the berth allocation problem only. Whereas approaches based on column generation have been recently proposed by Mauri et al. (2008) and Buhrkal et al. (2011), it represents the first exact branch-and-price algorithm for the BAP. On the other hand, the quay crane assignment is easily solved by a general-purpose solver. For all the details concerning this experiment, including the BAP and QCAP formulations, we refer the reader to Vacca (2011).

Handling time estimation In the hierarchical approach, the handling time of vessels is assumed to be known in advance. In practice, the expected handling time is provided by terminal planners, who base their

estimations on quantitative data such as vessel's workload, average QC productivity, availability of transfer equipment, vessel's priority, as well as on their experience. In particular, some extra time can also be included in the estimation in order to guarantee more robustness and flexibility to the schedule.

In our experiment, among the available TBAP data, we are given the set of feasible quay crane assignment profiles defined for every vessel and known in advance; in particular, we know the duration in terms of working shifts of every feasible QC profile, expressed by the input parameter t_i^p .

In order to start the entire hierarchical optimization process, we produce two different estimations for the handling time, both motivated by the practice:

Scenario A for every vessel, the handling time is given by the longest feasible quay crane assignment profile;

Scenario B for mother vessels, the handling time is given by the shortest feasible quay crane assignment profile whereas for feeders, the handling time is given by the longest feasible quay crane assignment profile.

Scenario A is very conservative and somehow represents the worst-case scenario, when all vessels are serviced at the lowest rate. However, this handling time estimation may be useful to produce robust schedules. Scenario B can be considered more realistic, since mother vessels typically have higher priority than feeders. In particular, we expect the terminal to operate as fast as possible mother vessels in order to minimize their stay at the port. Both scenarios are realistic and reasonable in practice.

Comparative analysis We consider instances with 10 vessels and 3 berths over a time horizon of 4 days. Such instances are obtained by reducing the time horizon from seven to four working days, in order to add congestion to the problem; in order to maintain feasibility, vessels' time windows are also relaxed and the quay crane capacity increases from $Q = 8$ to $Q = 10$. For additional results considering a time horizon of one week, we refer the reader to Vacca (2011).

10 x 3 Inst.	Scenario A			Scenario B			Integrated TBAP				
	opt_sol	K	t(s)	opt_sol	K	t(s)	opt_sol	K	t(s)	%(A)	%(B)
H1_10_4d	x			x			777398	3	39	∞	∞
H1_20_4d	776331	3	10	x			779674	3	47	0.43%	∞
H1_30_4d	776365	3	13	x			782300	3	66	0.76%	∞
H2_10_4d	718900	3	8	x			722431	3	28	0.49%	∞
H2_20_4d	719987	3	11	x			724345	3	36	0.61%	∞
H2_30_4d	719701	3	13	x			725585	3	28	0.82%	∞
L1_10_4d	507422	3	5	508657	3	7	512533	2	4	1.01%	0.76%
L1_20_4d	507304	3	4	508505	3	6	512533	2	19	1.03%	0.79%
L1_30_4d	507339	3	4	508540	3	6	512991	2	10	1.11%	0.88%
L2_10_4d	553971	3	6	x			558750	2	16	0.86%	∞
L2_20_4d	554380	3	6	556272	3	4	558786	2	25	0.79%	0.45%
L2_30_4d	554380	3	6	556280	3	4	558822	2	6	0.80%	0.46%

Table 10: *Optimal solutions for 10 vessels and 3 berths over 4 days.*

Table 10 compares the optimal solutions for the hierarchical approach under scenarios A and B to the integrated TBAP approach. For all solutions we report the value of the objective function (opt_{sol}), the number of used berths (K) and the computational time in seconds ($t(s)$). Columns ‘%(A)’, ‘%(B)’ indicate the improvement of the integrated solution with respect to the hierarchical approach under scenarios A, B respectively.

For congested instances the hierarchical approach clearly shows its drawbacks: first of all, it is not always able to provide a feasible solution. More specifically, the quay crane assignment may not be feasible for a given berth allocation plan, due to the QC capacity constraint; this is often the case for scenario B, where the shortest handling time is assigned to mother vessels, and therefore a higher number of cranes is used. In particular, the hierarchical approach fails in providing a solution for all the high-load instances (H1 and H2) under scenario B. On the contrary, the integrated approach finds the optimal solution for all tested instances. Surprisingly, the computational effort required by the integrated approach (always less than one minute) is comparable to the hierarchical approach: optimal integrated solutions are therefore produced in a fast and efficient way. Also, it is interesting to notice that the integrated solution makes use,

in some cases, of one berth less than the solution provided by the hierarchical approach.

Summing up, computational results confirm the added value of integration in terms of cost reduction and efficient use of resources. The main outcome of the analysis is that the strong assumptions made by the sequential approach may prevent to find any feasible solution, whereas the integrated approach always finds the optimal one. This occurs especially for congested instances. Furthermore, the additional effort required to solve the integrated problem is moderate.

7 Conclusions

In this paper we have presented a new model and a branch-and-price algorithm for the Tactical Berth Allocation Problem.

The model is based on an exponential number of variables and it is solved via column generation. In order to obtain integer solutions, a branch-and-price scheme has been implemented, introducing several accelerating techniques specifically designed for our problem.

Computational tests prove that our exact algorithm outperforms commercial solvers: especially on small instances, branch-and-price always provides optimal solutions relatively fast. The problem size still represents an issue and additional advanced techniques should be further investigated to overcome the complexity of the problem.

Furthermore, the proposed branch-and-price algorithm enables us to provide an experimental comparison between the traditional hierarchical approach (that sequentially solves the berth allocation and the quay crane assignment) and the integrated planning approach. Computational experiments confirm the added value of integration in terms of cost reduction and efficient use of resources.

Most of the presented accelerating techniques concern the pricing problem and prove to be very successful, reducing the computational time in the pricing by 90% on average. Future research should focus on improving the master problem. Alternative linearizations of the quadratic objective function should be investigated. Furthermore, cutting planes for the master problem should be studied in order to improve the bounding phase throughout the search tree.

References

- Addis, B., Carello, G. and Ceselli, A. (2009). A branch-and-price-and-cut approach for a two-level hierarchical location problem, *Proceedings of the International Network Optimization Conference (INOC)*.
- Barnhart, C., Belobaba, P. and Odoni, A. R. (2003). Applications of operations research in the air transport industry, *Transportation Science* **37**(4): 368–391.
- Beasley, J. E. and Christofides, N. (1989). An algorithm for the resource constrained shortest path problem, *Networks* **19**: 379–394.
- Bierwirth, C. and Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals, *European Journal of Operational Research* **202**(3): 615–627.
- Buhrkal, K., Zuglian, S., Ropke, S., Larsen, J. and Lusby, R. (2011). Models for the discrete berth allocation problem: A computational comparison, *Transportation Research Part E* **47**(4): 461 – 473.
- Choo, S., Klabjan, D. and Simchi-Levi, D. (2010). Multiship crane sequencing with yard congestion constraints, *Transportation Science* **44**(1): 98–115.
- Christiansen, M., Fagerholt, K. and Ronen, D. (2004). Ship routing and scheduling: status and perspectives, *Transportation Science* **38**: 1–18.
- Cordeau, J. F., Laporte, G., Legato, P. and Moccia, L. (2005). Models and tabu search heuristics for the berth-allocation problem, *Transportation Science* **39**(4): 526–538.
- Dell’Amico, M., Righini, G. and Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection, *Transportation Science* **40**(2): 235–247.
- du Merle, O., Villeneuve, D., Desrosiers, J. and Hansen, P. (1999). Stabilized column generation, *Discrete Mathematics* **194**: 229–237.
- Gélinas, E., Desrochers, G., Desrosiers, J. and Solomon, M. (1995). A new branching strategy for time constrained routing problems with application to backhauling, *Annals of Operations Research* **61**: 91–109.

- Giallombardo, G., Moccia, L., Salani, M. and Vacca, I. (2010). Modeling and solving the tactical berth allocation problem, *Transportation Research Part B* **44**(2): 232–245.
- Grønhaug, R., Christiansen, M., Desaulniers, G. and Desrosiers, J. (2010). A branch-and-price method for a liquefied natural gas inventory routing problem, *Transportation Science* **44**(3): 400–415.
- Imai, A., Chen, H. C., Nishimura, E. and Papadimitriou, S. (2008). The simultaneous berth and quay crane allocation problem, *Transportation Research Part E* **44**(5): 900–920.
- Imai, A., Nishimura, E. and Papadimitriou, S. (2001). The dynamic berth allocation problem for a container port, *Transportation Research Part B* **35**(4): 401–417.
- ISL (2009). *Shipping Statistics and Market review (SSMR) - Short Comment*, Vol. 53, Institute of Shipping Economics and Logistics, <http://www.isl.org>.
- Liberatore, F., Righini, G. and Salani, M. (2011). A column generation algorithm for the vehicle routing problem with soft time windows, *4OR: A Quarterly Journal of Operations Research* **9**: 49–82.
- Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation, *Operations Research* **53**(6): 1007–1023.
- Mauri, G., Oliveira, A. and Lorena, L. (2008). A hybrid column generation approach for the berth allocation problem, in J. van Hemert and C. Cotta (eds), *Evolutionary Computation in Combinatorial Optimization*, Vol. 4972 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 110–122.
- Meisel, F. and Bierwirth, C. (2009). Heuristics for the integration of crane productivity in the berth allocation problem, *Transportation Research Part E* **45**(1): 196–209.
- Park, Y. M. and Kim, K. H. (2003). A scheduling method for berth and quay cranes, *OR Spectrum* **25**(1): 1–23.

- Righini, G. and Salani, M. (2006). Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints, *Discrete Optimization* **3**(3): 255–273.
- Righini, G. and Salani, M. (2008). New dynamic programming algorithms for the resource constrained shortest path problem., *Networks* **51**(3): 155–170.
- Rousseau, L.-M., Gendreau, M. and Feillet, D. (2007). Interior point stabilization for column generation, *Operations Research Letters* **35**(5): 660 – 668.
- Stahlbock, R. and Voss, S. (2008). Operations research at container terminals: a literature update, *OR Spectrum* **30**(1): 1–52.
- Steenken, D., Voss, S. and Stahlbock, R. (2004). Container terminal operation and operations research - a classification and literature review, *OR Spectrum* **26**(1): 3–49.
- UNCTAD (2009). *Review of Maritime Transport*, United Nations Conference on Trade and Development, <http://www.unctad.org>.
- Vacca, I. (2011). *Container terminal management: integrated models and large-scale optimization algorithms*, PhD thesis, Ecole Polytechnique Fédérale de Lausanne.
- Vacca, I., Bierlaire, M. and Salani, M. (2007). Optimization at container terminals: Status, trends and perspectives, *Proceedings of the 7th Swiss Transport Research Conference (STRC)*.
- Vacca, I., Salani, M. and Bierlaire, M. (2010). Optimization of operations in container terminals: hierarchical vs integrated approaches, *Proceedings of the 10th Swiss Transport Research Conference (STRC)*.
- Vis, I. F. A. and de Koster, R. (2003). Transshipment of containers at a container terminal: An overview, *European Journal of Operational Research* **147**(1): 1–16.