The Berth Allocation and Quay Crane Assignment Problem Using a CP Approach

Stéphane Zampelli[1], Yannis Vergados[2], Rowan Van Schaeren[3], Wout Dullaert[4,5], and Birger Raa[3,6]

[1] Dynamic Decision Technologies, Providence
[2] Ph.D. Graduate from Brown University
[3] Antwerp Maritime Academy
[4] VU Amsterdam, Department of Informatics, Logistics and Innovation
[5] University of Antwerp, Institute of Transport and Maritime Management Antwerp
[6] Ghent University

Abstract. This paper considers the combination of berth and crane allocation problems in container terminals. We propose a novel approach based on constraint programming which is able to model many realistic operational constraints. The costs for berth allocation, crane allocation, time windows, breaks and transition times during gang movements are optimized simultaneously. The model is based on a resource view where gangs are consumed by vessel activities. Side constraints are added independently around this core model. The model is richer than the state of the art in the operations research community. Experiments show that the model produces solutions with a cost gap of 1/10 (7,8%) to 1/5 (18,8%) compared to an ideal operational setting where operational constraints are ignored.

Keywords: berth allocation, crane assignment, containers, terminal, constraint programming.

1 Introduction

A container terminal is a facility where cargo containers are transshipped between vessels and external trucks or trains. Cranes along the quay called 'quay cranes' are responsible for charging and discharging containers. Special trucks in the field move containers from quay cranes to container blocks in the yard. External trucks bring containers to the terminal and take away containers from the yard blocks. Many logistic problems arise in this context. We focus on two of them. On one hand, the berth allocation problem (BAP) positions vessels optimally, ensures security distances, and minimizes stay durations along the quay using a simplified model of the crane assignments. On the other hand, the quay crane assignment problem (CAP) considers the problem of detailed assignment of quay cranes to vessels in order to handle the incoming and outgoing containers where a feasible berth plan is already available. The challenge, proposed by our industrial partner, is to incorporate both problems together with those real-world constraints.

C. Schulte (Ed.): CP 2013, LNCS 8124, pp. 880–896, 2013.

Berth Allocation Problem. The BAP problem schedules the vessels by deciding the position of the vessel along the quay, estimating the duration needed to handle all the loading and discharging containers of the vessel, and avoiding vessel overlap along the quay. The difference with the CAP is that the stay duration along the quay is simplified by avoiding to compute the detailed crane assignment scheduling on each vessel. We review the detailed BAP problem constraints.

- The total length of all the vessels should be shorter than the quay length.
- Positions along the quay are represented by discrete bollards. The mooring ropes and wires used for securing the vessel along the quay length are attached to bollards. Every vessel is assigned a mooring place or berth that is a multiple of bollard distances. The distance between two bollards on the same quay length is equal. In Figure 1, the vessel uses bollard 2 to 5.
- Vessels along the quay should not overlap.
- The ideal berth of the vessel along the quay is computed by another yard optimization tool and is outside the scope of this paper. Ideal berth positions are an input in the context of this paper. Vessels discharge and load containers to and from containers blocks in the yard according to the yard planning. An ideal berth position can be precomputed for each vessel. The customer pays a fixed price for the container loading/unloading regardless of the yard position the container will occupy. The terminal wants to minimize the distance between the ideal berthing position and the precomputed one. Figure 2 represents a bad berth allocation.
- The computation of the handling time of the vessel depends on the cranes. In the BAP, this computation is simplified, not considering the detailed scheduling of the CAP. The handling time links the BAP and the CAP. Such a detailed scheduling is considered in the CAP below.
- Vessels have setup times. When a vessel arrives at a terminal and is safely moored alongside, the cranes can not immediately start to discharge the containers. The securing of the containers, called lashings, first need to be undone. The time needed for unlashing the containers differs per vessel and per stowage configuration. This time needs to be taken into account concerning the starting time for the cranes.

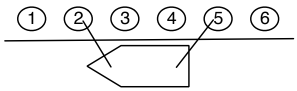

Fig. 1. Using bollards for defining the quay length occupied by a vessel

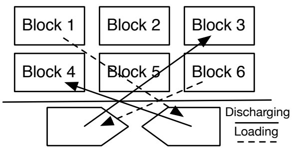

Fig. 2. A bad example of berth allocation regarding the yard distance cost

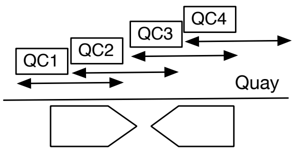

Fig. 3. An example of quay crane (QC) ranges on a container terminal with four quay cranes

Crane Assignment Problem. The CAP problem considers the detailed assignment of gangs to quay cranes and cranes to vessels. A gang is a team of workers consisting of a crane driver, a foreman, a person checking the container ids, two dockers to handle the containers and the driver moving the internal truck. Hiring a gang has a fixed cost per shift. Gangs have to be assigned to a crane but can be moved freely from one crane to another. A quay crane handles the containers to charge or discharge from or to a vessel. There are different types of cranes (Panamax, Post Panamax, Super-Post Panamax STS, ...). Vessels may accept only specific types of cranes. They also have a fixed arrival time at the terminal and leave as soon as all containers have been handled. A quay crane can be moved at any point in time from one vessel to another, creating a preemptive schedule. The overall goal is to minimize the operational cost of the terminal seen as a service from the point of view of the vessel operators. Let us review the detailed problem constraints.

- There is a maximum number of cranes available along the quay.
- Each vessel has a time window during which it needs to be handled alongside the quay length. The terminal operator will have to pay a fine, if the vessel arrives on schedule but cannot be handled within the agreed time frame.
- Repositioning cranes from one vessel to another takes 30 minutes.
- Gangs have breaks. Each gang works for eight hours. Each gang has a break of half an hour each four hours. During this break, crane repositioning is free, handled by a specialized team.
- The maximum number of cranes for each vessel is limited by the length and the number of bays of the vessel. Each quay crane has a fixed width, and hence a maximal number of cranes can work on one vessel simultaneously.
- Cranes operate on a common rail and have operating ranges. Cranes are electrically driven. The length of the source cables are chosen in such a way that an optimal coverage is given for the quay length (see Figure 3).
- Cranes are operated on a single rail, so they cannot cross each other.
- The gang cost per shift depends on the shift on which the gang operates. An example of relative gang costs is shown in Table 1. Note that gang costs must be paid in full, even if it only works during a part of a shift. There is always an integer number of gangs per shift.
- Crane productivity is measured in containers per hour and depends on a number of external factors (weather, crane driver, traffic, stowage plan, security vessel specific rule); however we consider the crane productivity as constant for all cranes along the horizon.

Table 1. Relative cost of a gang

	Weekday	Saturday	Sunday
Morning	1.05	1.50	2
Afternoon	1.15	1.50	2
Night	1.50	2	2

Table 2. Gang shifts and breaks

	Break 1	Break 2
Morning (06:00-14:00)	09:30-10:00	13:30-14:00
Afternoon (14:00-22:00)	17:30-18:00	21:30-22:00
Night (22:00-06:00)	01:30-02:00	05:30-06:00

Previous Work. We focus on literature about integrated BAP and CAP problems (BAPCAP). Readers interested in the abundant literature about BAP or CAP can refer to [4]. BAPCAP was first studied in [20]. They propose a two step approach. The first phase determines the berthing time and position of each vessel and the number of cranes to be allocated to the vessel. The second phase schedules the assignment of individual cranes, solved by a Lagrangian relaxation and dynamic programming respectively. Time is discretized in blocks of one hour. There is no detailed crane reallocation. Moreover, there is no consideration for gangs and gang cost. [19] gives a formulation of the BAPCAP, and solves the model in a two step procedures: a GA procedure (crossover and mutation) to assign ships in order to berth, and a heuristic to assign cranes to ships. [18] solves the BAPCAP by using a 4 steps GA-based approach, successively locating ship to berths, assigning quay cranes to berths, and designing berth and quay cranes scheduling. Note that there is no consideration for gang cost and cranes cannot be re-allocated. [17] decides on the berthing position, the berthing time, and the number of cranes that serve a vessel within the handling period, by taking into account drop of crane productivity due to interference. The BAPCAP model is then solved using heuristics and metaheuristics. Time is discretized in blocks of one hour. Detailed crane assignment and reallocation is not considered and authors suggest it should be post-processed. [16] proposes to optimize the cranes efficiency. The block periods consist of 12h and the horizon is limited to 6 blocks (3 days) because of the model complexity. It is extended by using a rolling horizon technique [7]. Among recent works, [15] considers many detailed constraints and studies the BAPCAP under uncertainty. Limited crane reallocation when a new vessel arrives can occur. The wharf is also segmented into fixed length segments. A non linear mixed integer model is solved using GA. Gang, shift costs and breaks are ignored. [6] proposes a pure MIP BAPCAP approach with time bucketized in periods of 2 hours and a rolling horizon. Recently, [14] splits the problem in two, with BAP on one side solved by GA and CAP is solved by a mixed-integer linear program. A Bi-Level Programming (BLP) approach is used to combine both subproblems. Running time is pretty high with 480 minutes reported for 3 vessels and 8 QCs, although the detailed scheduling of the crane is taken into account. [13] solves the two problems independently, although the BAP is continuous, meaning the quay length is not discretized in blocks. They use a nested loop-based evolutionary algorithm (NLEA), and two inner loops and one outer loop are suggested. Let us stress that in the OR literature, no papers can handle the set of constraints proposed in this paper in a single model. Our model schedules to the minute with dynamic crane reallocation and a computation of the actual gang and shift costs.

Paper Main Contributions. Paper contributions are threefold:

– The solved problem tackles a combination of many complicated technical constraints, such as setup times, transition times, time windows, shifts, breaks, smooth workforce allocation (work for large consecutive spans of time), spatial positioning, etc., in a large scale and realistic setting (5 days,

30 vessels). The crane allocation itself is done by a tractable subset inside the main model. It shows how useful CP is in tackling those OR problems.
– The proposed model shows how to solve a complex problems not only by identifying the main underlying structure in a form of a constraint but also by isolating several aspects into submodels and making them communicate through the variables. This is an interesting approach to tackle other challenging OR problems.
– This paper shows a case where CP can bring benefits to challenging problems inside the OR community that tends to be MIP or heuristics dominated. It has been agreed for a long time that combining detailed quay crane scheduling and berth planning is not an easy problem. MIP models become complex and heuristics are used to overcome this issue.

Benefits of CP. The scheduling features of CP are one of the keys against MIP or heuristics centric approaches as scheduling the cranes is the core of the problem. The declarativeness of CP was important, since more than 20 different models were tested, in a reasonable development cycle time (<2 men/month). It would have been difficult to test all those ideas using a heuristics approach, given the amount of coding and testing required. Our industrial partner did not believe all those constraints could be handled in a declarative way. CP declarativeness allowed not only to identify and state constraints but also to identify and integrate submodels of the problem at hand. The ability to easily add small side constraints also played a key role. For instance it is easy for a user to state in the model that a specific crane will go down in a given period of time or forbid certain crane/vessel assignments because of compatibility issues.

2 Model Description

Our CP model combines several submodels. The core model, described in Section 2.1, allocates gangs to vessel subtasks, minimizing the total gang cost and the lateness. The crane allocation and the berthing are added to the core model, in Section 2.2 and 2.3. Section 2.4 describes the objective function. A labeling procedure is proposed in Section 2.5. Section 3 discusses computational results.

2.1 Gang Allocation

Gangs need to be allocated to cranes, and cranes to vessels. Gangs can move freely from crane to crane and cranes can be reassigned at any time. The key idea is to view the gangs as a resource and use cumulative constraints. Viewing the cranes as a cumulative resource is a deadend since cranes are ordered and have exotic constraints like range and non crossing constraints. Each vessel is a set of activities that consumes a number of gangs in a preemptive way. The actual crane assignment is left to a separate submodel that makes sure the assignment is possible. Each gang delivers a certain amount of workforce that depends on the duration linearly. This workforce idea makes it easier to compute the shift gang cost and deal with side constraints such as breaks and setup times.

Notations. A range R is a consecutive finite sequence of integers; its minimum (maximum) is noted $\underline{R}$ (resp. $\overline{R}$). The range of input vessels is denoted $vessels$, and for each $b \in vessels$, the range of vessel activities is denoted Act_b. The time horizon $Horizon$ is a range of time units of 1 minute. The range $Shifts$ indexes the shifts. The shift duration (including breaks) is noted sd. The range $Gangs$ indexes the gangs. The ranges $Gangs_b = [0, mc_b]$ with $b \in vessels$ are the possible values for the number of cranes that can be allocated to a vessel. The ranges $Breaks$ is the ranges of breaks. We assume those ranges start at zero. The lower bound (resp. upper bound) of a finite domain variable x is denoted $\underline{x}$ (resp. $\overline{x}$).

Before declaring activities and constraints, we convert containers to the concept of workforce. A unit of workforce is the work of one gang during one unit of time (1 minute). This conversion is needed because the scheduling of gangs activities over vessels depend on gang units and time units, and know nothing about containers. Workforce is the link between the scheduling of gangs and the containers of vessel. Each vessel needs a minimum amount of workforce to leave. The following two definitions grasp those ideas:

Definition 1 (Crane Productivity). *The productivity of a crane is the number of containers it can handle per hour.*

Definition 2 (Workforce). *Given a crane productivity p, the workforce needed to handle c containers is defined by* $(c*60)/p$. *The required workforce of a vessel, noted* mw_b*, is the workforce corresponding to its number of containers to handle.*

The only drawback is that a crane may be reassigned while a container is being moved, since only the required time is considered. However, this limitation has no impact on real operations: transition times can be shortened or extended to handle those cases in practice. We now define the set of activities $a_{b,i}$ with $b \in vessels$ and $i \in Act_b$.

Definition 3 (Activity). *An activity* $a_{b,i}$ *is defined by five variables:*

- $s_{b,i}$ *is the starting time,*
- $e_{b,i}$ *is the completion time,*
- $d_{b,i} = e_{b,i} - s_{b,i}$ *is the duration,*
- $cap_{b,i}$ *is the number of resources consumed by the activity between its starting time and its completion time.*
- *and* $wkf_{b,i}$ *is the workforce delivered by the task, with* $0 \leq wkf_{b,i} \leq cap_{b,i} * d_{b,i}$.

Our model creates one activity $a_{b,i}$ per vessel b and per index $i \in Act_b$. The capacity $cap_{b,i}$ is the constant number of gangs used by the activity. The equality of $wkf_{b,i}$ with $cap_{b,i}*d_{b,i}$ is not enforced because of breaks and transition times. For instance, if an activity overlaps with a break, the delivered workforce is inferior to its surface. An activity is an allocation of workforce to a vessel. Breaks and transition times are handled at the end of this section. Activities can be interrupted and are also optional (they can have a zero duration). Each vessel has its own time window. In the following, we abuse notations and use i instead of (b, i) when it is clear from context that we are speaking about a given boat.

Definition 4 (Time Window). *The time window of a vessel* $b \in vessels$ *is the couple* (ta_b, td_b)*, where the integer* ta_b *denotes the arrival time of the vessel* b *and* td_b *the deadline of vessel* b*.*

Arrival time for each vessel $b \in vessels$ and each index $i \in Act_b$ is enforced:

Constraint 1 (Arrival). $\forall\, b \in vessels, i \in Act_b : s_{b,i} \geq ta_b$

Constraint 2 (Required Workforce). $\forall\, b \in vessels : \sum_{i \in Act_b} wkf_{b,i} \geq mw_b$

Let us ignore shifts for now. At any point in time, there is maximum $\overline{Gangs}$ gangs that can be hired. Given two variables s and d representing the starting time and the duration variables of an activity a_i, the mandatory part noted $mand(a_i)$ or $mand(s, d)$ is a range $[\overline{e}-\underline{d}, \underline{s}+\underline{d}]$ that can be empty if the mandatory range does not exist. This can be modeled by a cumulative constraint:

Definition 5 (Cumulative). *Consider a resource limited by a constant capacity* c*, and a set of activities* $a_j \in A$*. A constraint* $cumulative(\{a_j \mid j \in A\}, c)$ *ensures the following constraint:* $\forall\, t \in Horizon \sum_{j \in I} cap_j \leq c$ *where* $I = \{j \in A \mid t \in mand(a_j)\}$*.*

Activities may not exceed the maximum number of available gangs:

Constraint 3 (Global Cumulative). *The following constraint is added to the model:* $cumulative(A, \overline{Gangs})$ *where* A *is the set* $\{a_{b,i} \mid b \in vessels,\ i \in Act_b\}$*.*

Each vessel is also constrained on its maximum number of gangs at any point in time. An additional $|vessels|$ number of cumulative constraints are posted:

Constraint 4 (Local Cumulative). *For each* $b \in vessels$*, the following constraint is posted:* $cumulative(A, \overline{Gangs_b})$ *where* A *is the set* $\{a_{b,i} \mid b \in vessels,\ i \in Act_b\}$ *and* $Gangs_b$ *is the possible gang range for vessel* b*.*

Let us introduce shifts in the model. For each shift, a variable denoting the number of gangs used can be created:

Definition 6 (Gang Shift). *For all* $sh \in Shifts$*,* $nbGangs_{sh}$ *is the number of gangs used in shift* sh*.*

For each shift, a fake activity is created that spans over the whole shift and consumes the number of gangs that are not used during that shift.

Definition 7 (Fake Activities). *For all* $sh \in Shifts$*, a fake activity* fa_{sh} *is created with the following domains:*

- *starting time* $s_{sh} = sh * sd$
- *duration* $d_{sh} = sd$

- *capacity* $cap_{sh} = \overline{Gangs} - nbGangs_{sh}$
- *workforce* $w_{sh} = 0$.

Let us introduce breaks and transition time. Two break intervals are present in each shift sh, a first break

$$[\frac{se_{sh}}{2} - bd, \frac{se_{sh}}{2}]$$

and a second break:

$$[se_{sh} - bd, se_{sh}]$$

where se_{sh} is the ending time of the shift sh and and bd is the constant break duration. Each break $r \in Breaks$ can be associated with such an interval noted b_r. A variable bi_r is equal to time intersection between b_r and $[s_{b,i}, e_{b,i}]$. The total intersection between an activity and the breaks can be measured:

$$bi_{b,i} = \sum_{r \in Breaks} bi_r \ .$$

Regarding transition times, we considered a fixed and constant transition time denoted $transitionTime$ that is assigned to all activities. The transition time can be defined as

$$tt_{b,i} = max(0, transitionTime - fb_{b,i})$$

where $fb_{b,i}$ is defined as:

$$\begin{aligned} fb_{b,i} &= bi_r \ \ \text{where } r = min\{r \in Breaks \mid bi_r \neq 0 \wedge s_{b,i} \in b_r\} \\ &= \ 0 \quad \text{if } r \text{ does not exist.} \end{aligned}$$

The variable $fb_{b,i}$ denotes the intersection of a break with the beginning of a vessel operation. Indeed, cranes can be moved during breaks. Breaks occurring at the beginning of vessel operations hence shorten transition time. The actual workforce of the activity (b, i) can be defined.

Constraint 5 (Workforce). *For each activity* (b, i)*, the workforce is*

$$wkf_{b,i} = (d_{b,i} - bi_{b,i} - tt_{b,i}) * cap_{b,i} \ .$$

Regarding the setup time, the transition time assigned to the first activity of the vessel stands for both the transition time of the cranes and the setup time. In this core model, gangs are assigned to vessels, using preemptive activities. Breaks and transition times are taken into account using the workforce variables. This first model is a relaxation of the problem as actual cranes along the quay are not assigned to vessels and vessel conflicting positions are ignored.

2.2 Space Allocation

Along the quay, the vessels should not overlap. The length of a vessel b is noted $length_b$. Let us define a vessel position along the quay:

Definition 8 (Position). *The position of vessel b along the quay is a finite domain variable and is denoted pos_b.*

Let us define the starting and ending time of vessel:

Definition 9 (Vessel Time Window). *The starting time of a vessel b is $s_b = \min_{i \in Act_b} s_{b,i}$, and its ending time is $e_b = \max_{i \in Act_b} e_{b,i}$.*

Non overlap between vessels is stated by enforcing that vessels overlapping in time should not overlap in space:

Constraint 6 (Non-overlap). $\forall\ (b, c) \in vessels \times vessels, b > c : (s_b \leq e_c \wedge e_b \geq s_c) \vee (s_c \leq e_b \wedge e_c \geq s_b) \Rightarrow (pos_c \geq pos_b + length_b) \vee (pos_b \geq pos_c + length_c)$

2.3 Crane Allocation

In this section a tractable submodel is presented for the crane allocation. This model can filter any inconsistent crane assignment value once the information is available from other submodels.

The first concept is the crane range. The assignment of cranes to a vessel can be represented as a range, because all cranes are consecutive along the quay and cannot cross each other, since they are each operated on a single rail.

Definition 10 (Crane Range). *The crane range of an activity (b, i) ($i \in Act_b$) is a range $[sc_{b,i}, ec_{b,i}]$, where $sc_{b,i}$ is the starting crane and $ec_{b,i}$ the ending crane. The variable $nbCranes_{b,i}$ denotes the number of cranes assigned to vessel activity (b, i).*

The following constraint holds: $sc_{b,i} \leq ec_{b,i}$, and the number of cranes and the crane range are linked: $nbCranes_{b,i} = ec_{b,i} - sc_{b,i} + 1$.

Each crane has a certain span along the quay, because of physical constraints. This means that a crane can be assigned to a vessel if and only if the crane can reach the vessel along the quay. Given a vessel b, only a subset of crane ranges are available for vessel b. Let us define the $craneMin$ array indexed by bollard positions. Since we focus on a given boat, we omit subscripts. The value $craneMin_p$ (resp. $craneMax_p$) is the leftmost crane (resp. rightmost crane) that can reach bollard range $[p, p + length_b]$. The consistency between crane positions and vessel positions can be added to the model:

Constraint 7 (Crane Position). $\forall\ b \in vessels, i \in Act_b : sc_{b,i} \geq craneMin[pos_b]$ *and* $ec_{b,i} \leq craneMax[pos_b]$.

The following set of constraints distribute the cranes between activities.

Constraint 8 (Crane Allocation). *For each pair of distinct tasks* $((b,i)$, $((c,j))$ *overlapping in time, their crane range follows their relative position:*

$$[(s_{b,i} \leq e_{c,j} \wedge e_{b,i} \geq s_{c,j}) \vee (s_{c,j} \leq e_{b,i} \wedge e_{c,j} \geq s_{b,i}) \wedge \ (pos_b < pos_c)] \Rightarrow ec_{b,i} < sc_{c,j}$$

and:

$$[(s_{b,i} \leq e_{c,j} \wedge e_{b,i} \geq s_{c,j}) \vee (s_{c,j} \leq e_{b,i} \wedge e_{c,j} \geq s_{b,i}) \wedge \ (pos_b > pos_c)] \Rightarrow sc_{b,i} > ec_{c,j}.$$

Once the position, the time span and the number of cranes of pairwise activities are bound, the right side constraints from Constraint 8 form a linear chain of inequality constraints. Given a time $t \in Horizon$, a total order is enforced upon crane range variables of activities intersecting in time t. Ignoring distinction between vessel and activity indexes, we have at a given time $t \in Horizon$:

$$sc_1 \leq_{k1} ec_1 < sc_2 \leq_{k_2} ec_2 < \ \ldots \ \leq_{k_{n-1}} ec_{n-1} < sc_n \leq_{k_n} ec_n \qquad (A)$$

where n is the number of vessel activities intersecting in time with t. $\leq_{k_i}$ is a notation for the binary constraint $s_i \leq e_i - k_i + 1$, k_i is the bound value of variable $nbCranes_i$, and $<$ is the binary inequality constraint.

It is well-known [Jeavons, 1995] that max-closed (or min-closed) constraints and arc-consistency detect at fixpoint if a constraint system is satisfiable. Both constraints $x < y$ and $x \leq_k y$ are max-closed and min-closed[1]. This implies the following property:

Property 1. Suppose the arc-consistent fixpoint has been computed for the chain of constraints (A) and the fixpoint does not fail. Then any value from any variable in the set of variables of (A) can be extended to a solution.

This last property implies that the labeling of the crane range variables can be skipped as propagation will ensure crane ranges can be instantiated.

2.4 Objective

The three components of the objective includes the lateness cost, cost induced by the distance with the ideal position, and the total gang cost. The lateness of a vessel $b \in vessels$ is easily defined:

Definition 11 (Lateness). *The lateness* l_b *of a vessel* $b \in vessels$ *is equal to* $\max(0, e_b - ta_b)$.

Lateness is the exceeded handling time with respect to the deadline of the vessel time window. Let pos_b be the position variable of vessel b. A position difference can be defined similarly:

Definition 12 (Distance Gap). *The distance gap* dp_b *of a vessel* $b \in vessels$ *with respect to its ideal position* ip_b *is equal to* $|ip_b - pos_b|$.

[1] We omit the proof due to lack of space. See [23] for the full proof.

The number of gangs used in each shift is defined by $nbGangs_{sh}$, see Section 2.1.

Constraint 9. *The objective variable obj is defined as*

$$obj = \sum_{b \in vessels} (l_b * lc_b) + \sum_{b \in vessels} (dp_b * dc_b) + \sum_{sh \in Shifts} (nbGangs_{sh} * gc_{sh})$$

where lc_b is the lateness cost per minute for vessel b, dc_b is the distance cost per meter for vessel b, and gc_{sh} is the cost of a single gang in the shift sh.

2.5 Labeling

The primary goal of the labeling is to minimize the total gang cost per shift while avoiding lateness. When the minimization of a resource is required in the cumulative constraint, a *fill hole* heuristic is used. The idea is to fill holes present inside the profile of the resource usage. A similar labeling has been used in the context of a soft cumulative [22]. The profile of a cumulative constraint can be defined as:

Definition 13 (Profile). *The profile of a cumulative constraint is a set of tuples (t_i, d_i, v_i), $i \in P$, such that:*

- *(non-overlap)* $\forall\, i, j \in P,\ i \neq j : [t_i, t_i + d_i - 1] \cap [t_j, t_j + d_j - 1] = \emptyset$
- *(usage reflection)* $\forall\ t \in Horizon\ \exists\ i \in P : \sum_{k \in A} cap_k = v_i$ *where* $t \in [t_i, t_i + d_i - 1]$ *and* $A = \{j \in Act \mid t \in mand(a_j)\}$
- *(cover)* $\forall\, t \in Horizon\ \exists\, i \in P : t \in [t_i, t_i + d_i - 1]$

The set *Act* denotes the set of all activities. Tuples of a profile are called *segments*.

Definition 14 (Minimal Profile). *A cumulative profile is minimal iff* $\forall\, i, j \in P, i \neq j$, $v_i \neq v_j$, *that is* $|P|$ *is minimal.*

In the following, we shall suppose that P is ordered with respect to t_i. We note invariably $i \in P$ and $(t_i, d_i, v_i) \in P$. Holes are defined with respect to left and right segments. The left (right) segment i of a profile P is the segment $i - 1$ (resp. $i + 1$). Its left (right) segment value is v_{i-1} (resp. v_{i+1}). The left and right segment of i may be undefined if $i = \min(P)$ or $i = \max(P)$. If they are undefined, their left or right segment value is equal to $\overline{Gangs}$.

A hole is an augmented segment. The profile segment is augmented with a depth information h:

$$h = \begin{cases} min(l - v_i, r - v_i) & \text{if } l - v_i > 0 \text{ and } r - v_i > 0 \\ l - v_i & \text{if } l - v_i > 0 \text{ and } r - v_i < 0 \\ r - v_i & \text{if } l - v_i < 0 \text{ and } r - v_i > 0 \\ 0 & \text{if } l - v_i \leq 0 \text{ and } r - v_i \leq 0 \end{cases}$$

where l and r are the left segment value and the right segment value resp. We say a segment is augmented by its hole value h.

The heuristic function uses a function called lmdh() for *leftmost deepest hole*. It returns an ordered sequence of holes based on the profile of the cumulative constraint that the next activity should try to fill. More specifically, considering the minimal profile P of the cumulative constraint, it returns a sequence O of augmented segments (t_j, d_j, v_j, h_j) such that:

1. O defines for C the same profile as P:
   $\forall\ t \in Horizon\ \exists\ j \in O : \sum_{k \in A} cap_k = v_j$ where $A = \{k \in Act \mid t \in mand(a_k)\}$.
2. Segments of O cannot cross shift boundaries
   $\forall\, j \in O, \exists\, sh \in Shifts : t_j \geq sh * sd \wedge t_j + d_j - 1 \leq ((sh+1) * sd) - 1$.
3. h_j is the augmented hole value from the segment $i \in P$ for which $v_i = v_j$
4. the sequence O is sorted lexicographically on highest h_i and smallest t_i.

In other words, $lmdh()$ returns the same segments as P, except they are split at any shift beginning and they are ordered.

The labeling procedure is described in Algorithm 1. The vessels are scanned in increasing arrival time ta_b (line 1) and the activities of vessel b are scanned (line 3). The amount of workforce still to deliver is computed (line 4), and if no workforce is left, the remaining activities Act_b are assigned to a duration of zero so that they do not appear in the solution (line 4 to 7). If there is some work to do on the current vessel, the profile holes are then computed based on the information of the cumulative constraint, by calling lmdh() (line 8). The holes are ordered according to the gang cost corresponding to the shift they are in. The selected activity is forced to be included into the width of hole (line 9 to 11). The depth of the hole is adjusted if it is a border case. This can happen for instance if the left segment is undefined. Another possibility is that $h = 0$ because the segment is a hill. In both cases, h is set to the maximum possible number of gangs for the activity (line 13 to 15). The number of gangs, based on the augmented segment, tend to be the number of gangs that would fill the hole vertically, if any. Then the number of gangs is assigned, the activity is pushed leftmost, and the workforce delivered is maximized, maximizing the width of the activity (line 17 and 19). The current index of the activity is added to already used activities (line 23). When all activities of current vessel have been scheduled, line 25 and 26 assign a position to the vessel along the quay. It should be stressed that the crane allocation range variables are not labeled, as the crane allocation submodel is tractable, see Section 2.3.

The above labeling obtains good solutions. Using a naive labeling, where activities are pushed leftmost lead to worse results as demonstrated in the experiments. Moreover, we use large neighborhood search [9], where entire vessels are fixed with a 0.6 probability.

3 Computational Results

This section measures the performance of the proposed model on generated datasets and on a industrial dataset. To the best of our knowledge, most terminals schedule crane and berth by hand. Academic papers cover too few real-world

PROCEDURE label()
for all $b \in vessels$ by arrival order **do**
  $I \leftarrow \emptyset$ //I is the set of activities already used
  for all $i \in Act_b : i \notin I$ **do**
    int $lw \leftarrow mw_b - \sum_{i \in A_b} \underline{wkf}_{b,i}$ //workload left
    if $lw \leq 0$ **then** //if nothing to do for this vessel
      try constraint $d_{b,i} = 0$ //impose zero duration, as this activity is not used
    else
      for all $[t_i, d_i, v_i, h_i] \in$ lmdh() in increasing shift cost order **do**
        $h_1 \leftarrow t_i$; $h_2 \leftarrow t_i + d_i - 1$;
        try constraint $s_{b,i} \geq h_1$ //restrict activity to the segment $[h_1, h_2]$
        try constraint $e_{b,i} < h_2$
        $h \leftarrow h_i$
        if $h_i = \underline{0 \text{ or } h_i >} \overline{nbCranes}_{b,i}$ **then** //if it is not a proper hole
          $h \leftarrow \overline{nbCranes}_{b,i}$ //set to max nbr of gangs for vessel b
        end if
        for all gangs g from h down to $\underline{nbCranes}_{b,i}$ **do**
          try constraint $nbCranes_{b,i} = g$ //impose nbr of cranes, starting from depth h
          try constraint $s_{b,i} = \underline{s_{b,i}}$
          try constraint $wkf_{b,i} = \overline{wkf}_{b,i}$ //fix duration, as start and nbr of gangs are fixed
        end for
      end for
    end if
    $I \leftarrow I \cup \{i\}$
  end for
  try constraint $diffPos_b = \underline{diffPos}_b$ //label position close to the ideal position
  try constraint $pos_b = \underline{pos}_b$ //diffPos is an absolute value
end for

Algorithm 1. Dedicated labeling for the global model.

constraints. Each previous work has its own set of constraints and a comparison would not be fair. Commercial tools do not optimize globally and are a help to build the schedule by hand. Additional details can be found in [23].

Datasets Description. In order to validate the model, we generated datasets based on the authors' experiences and information found in various published academic papers. Industrial datasets were also used.[2] The onset for generating our instances meet client's operational requirements. Vessels are planned in advance with a time horizon of 5 days (7200 minutes). The total quay length is 2000 meters, matching the largest container terminal in the world, and there are up to 30 vessels. The average crane productivity is 35 containers per hour or 0.5833 per minute. The total amount of quay cranes available is set to 19. Crane

[2] Industrial datasets are available upon request.

width is 80 meters. This means that a vessel of 230 meters e.g. would have at most 3 cranes working on it simultaneously: $\lfloor 230/80 \rfloor$. Bollards are 20 meters apart. This distance is also used to add to the vessel's length around the vessel for safe mooring alongside the quay length. If a vessel stays longer than allowed by its commercial time window, the lateness cost is 5000€ per hour. Deviation with the ideal berth position costs one euro per meter of deviation. The gang costs use Table 1 and a base cost of 2600€. Shift details (working hours and breaks) are shown in Table 2. Setup leaving and arriving times and transition times for cranes are set to 20 minutes. The set Act_b is an input. The model uses 1 activity for barges with less than 35 containers. For other vessels, the number of containers (or workforce) is divided by a split threshold, typically a workforce of 100 containers for 4 cranes. More activities are useless (0 workforce) below this threshold.

MIP Relaxation. We need a measure of the gap with respect to optimality. The client uses MIP and the optimality gap is an expected output. We relaxed the gang allocation core submodel (see Section 2.1) into an integer program. This relaxation gives a lower bound to measure a gap with respect to an ideal operational setting. Crane allocation and space allocation submodels are ignored. Cranes can reach any vessel, can cross each other and can move instantly. Vessels can overlap along the quay. The MIP model considers cranes are helicopters and vessels can be positioned anywhere. Considering all vessels, the required mw_b has to be distributed into legal shifts (shifts intersecting with their vessel time windows) so that the total gang cost is minimized. The proposed MIP model is a lower bound relaxation of the gang allocation model from Section 2.1. A detailed description of this MIP model can be found in [23].

Results. The goal of our experiments is to measure the optimality gap between the CP model and the relaxed MIP model. All runs were performed on a 2,53Ghz Intel CPU with 1GB of RAM with a timeout of 10 minutes. The MIP solver is SCIP [8] and the constraint programming solver is Comet.

Three models were used. All models use an LNS procedure that randomly fixes vessels with a 0.6 probability. The first one is the *fill-hole* model that uses the *fill hole* labeling, denoted FH. The second model is the *naive* model where a naive labeling is used to assign activities in a leftmost manner ignoring the profile. The last one is the *fill-hole-relax* model (denoted FHR) where there is no crane range constraints, no non-overlap constraints, no transition time and time windows are relaxed to the boundary of the shift. The line FHR solves a simplified core model to compare the MIP relaxation and the CP approach.

Table 3 shows the results. Both MIP and CP approaches have a timeout of 600 seconds. If the MIP time column displays a time less than 600 seconds, optimality has been proven by the MIP. The CP time column displays the time of the last solution found. The distance in percentage with the MIP objective value is given in column GAP. The four columns under 'Objective Value' denotes the total objective value, the gang cost, the position cost, and the lateness cost.

Table 3. Results for all instances

	Time (sec)		GAP	Objective Value				Extra Gangs
H	CP	MIP		Total	Gang	Pos.	L.	
Random1, 10 vessels								
FH	504	600	7.8	20648	20589	59	0	5(67/62)
naive	600	600	-	-	-	-	-	-
FHR	175	243	0.4	18522	18522	0	0	0(62/62)
Random2, 10 vessels								
FH	483	8	11.0	20553	20446	107	0	6(65/59)
naive	385	7	27.8	25356	25321	35	0	7(66/59)
FHR	93	6	0.4	18314	18314	0	0	0(59/59)
Random3, 10 vessels								
FH	542	343	18.8	36433	36265	168	0	12(104/92)
naive	600	356	-	-	-	-	-	-
FHR	364	600	0.7	28587	28587	0	0	0(92/92)
Random4, 10 vessels								
FH	582	600	13.6	29998	29473	525	0	6(86/80)
naive	600	600	-	-	-	-	-	-
FHR	211	600	0.4	26509	26509	0	0	0(80/80)
Industrial, 15 vessels								
FH	458	2	11.9	15857	15666	191	0	4(48/44)
naive	428	3	23.3	18209	18078	131	0	8(52/44)
FHR	501	2	0.9	14030	14030	0	0	0(44/44)
Industrial, 30 vessels								
FH	60	12	16.5	29884	29050	834	0	11(90/79)
naive	338	12	41.1	42335	41530	805	0	26(105/79)
FHR	12	11	1.8	25878	25878	0	0	1(80/79)

Finally, the number of additional gangs hired with respect to the lower bound MIP approach is printed in column 'Extra Gangs'. A line marked '-' means the constraint programming model did not find any solution before the timeout.

Naive labeling performs poorly compared to the *fill hole* labeling used by the *fill-hole* model. The *naive* model did not find any solution before the timeout in 3 out of 4 random instances and uses two times the number of gangs in the industrial instances. The *naive* model tends to have a lower position cost. The *fill-hole-relax* CP approach is trapped in local optima, but finds good solutions up to 2%. This is expected as MIP is known to be stronger for flow-like problems. The overall performance of our proposed approach is 1/10 (7,8%) to 1/5 (18,8%) of additional cost compared to an ideal operational world (the MIP lower bound).

4 Conclusion

Container terminals are more and more automated and as a result optimization technologies are needed to efficiently solve the numerous logistics problems arising. This is also reflected in the operations research literature where recent works try to solve these integrated problems. The question is whether CP can help in this quest. We answer this question by considering the integration of two problems using a real world constraints with an industrial partner.

We have shown that operational and realistic constraints for BAPCAP can be successfully addressed in the context of a CP approach. This approach is modular in the sense that each set of operational constraints can be separated. The key idea is to use the gang allocation process as the main component, and view it as a resource. Other side constraints can be integrated around this basic model. Experiments show that the CP model can produce solutions close to 1/5th to 1/10th from an ideal operational world. Overall, this work shows that CP can be a technology of choice for tackling challenging problems in the maritime industry considered "out of scope" for the current approaches, even under complex operational and scale constraints.

Future research includes using alternative profile-centered labeling or additional LNS procedures. The resource view of the model opens the possibility to use many scheduling tools from the OR/CP community to improve performance or to integrate new types of side constraints. Integrating the yard management aspect by computing the ideal positions together with the scheduling would extend the integrated approach, for which CP may be the right optimization technology.

References

1. Drewry Shipping Consultants Annual Container Market Review and Forecast 2007/08.A London, 250 p. (2007)
2. Drewry Shipping Consultants Container Forecaster 1Q08 London, 218 p. (2008)
3. Grossmann, H., Otto, A., Stiller, S., Wedemeier, J.: Growth Potential for Maritime Trade and Ports in Europe. Intereconomics, 226–232 (2007)
4. Bierwirth, C., Meisel, F.: A survey of berth allocation and quay crane scheduling problems in container terminals. European Journal of Operational Research 202, 615–627 (2010)
5. Moorthy, R., Teo, C.-P.: Berth management in container terminal: the template design problem. OR Spectrum 28, 495–518 (2006)
6. Raa, B., Dullaert, W., Van Schaeren, R.: An enriched model for the integrated berth allocation and quay crane assignment problem. Expert Systems with Applications 38(11), 14136–14147 (2011)
7. Zhang, C., Zheng, L., Zhang, Z., Shi, L., Armstrong, A.J.: The allocation of berths and quay cranes by using a sub-gradient optimization technique. Computers and Industrial Engineering 58, 40–50 (2010)
8. Achterberg, T., Berthold, T., Koch, T., Wolter, K.: Constraint Integer Programming: A New Approach to Integrate CP and MIP. In: Trick, M.A. (ed.) CPAIOR 2008. LNCS, vol. 5015, pp. 6–20. Springer, Heidelberg (2008)
9. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: Maher, M.J., Puget, J.-F. (eds.) CP 1998. LNCS, vol. 1520, pp. 417–431. Springer, Heidelberg (1998)
10. Daganzo, C.F.: The crane scheduling problem. Transportation Research Part B 23, 159–175 (1998)
11. Peterkofsky, R.I., Daganzo, C.F.: A branch and bound solution method for the crane scheduling problem. Transportation Research Part B 24, 159–172 (1990)
12. Liu, J., Wan, Y.-W., Wang, L.: Quay Crane Scheduling at Container Terminals To Minimize the Maximum Relative Tardiness of Vessel Departures. Naval Research Logistics 53, 60–74 (2006)
13. Yang, C., Wang, X., Li, Z.: An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. Computers & Industrial Engineering 63, 243–253 (2012)
14. Song, L., Cherrett, T., Guan, W.: Study on berth planning problem in a container seaport: Using an integrated programming approach. Computers & Industrial Engineering 62, 119–128 (2012)
15. Han, X., Lu, Z., Xi, L.: A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. European Journal of Operational Research 207, 1327–1340 (2010)
16. Chang, D., Jiang, Z., Yan, W., He, J.: Integrating berth allocation and quay crane assignments. Transportation Research Part E 46, 975–990 (2010)

17. Meisel, F., Bierwirth, C.: Heuristics for the integration of crane productivity in the berth allocation problem. Transportation Research Part E 45, 196–209 (2009)
18. Liang, C., Huang, Y., Yang, Y.: A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning. Computers & Industrial Engineering 56, 1021–1028 (2009)
19. Imai, A., Chen, H., Nishimura, E., Papadimitriou, S.: The simultaneous berth and quay crane allocation problem. Transportation Research Part E 44, 900–920 (2008)
20. Park, Y., Kim, K.: A scheduling method for Berth and Quay cranes. OR Spectrum 25, 1–23 (2003)
21. Zhang, C.Q., Liu, J.Y., Wan, Y.W., Murty, K.G.: Storage space allocation in container terminals. Transportation Research Part B 37, 883–903 (2003)
22. De Clercq, A., Petit, T., Beldiceanu, N., Jussien, N.: Filtering Algorithms for Discrete Cumulative Problems with Overloads of Resource. In: Lee, J. (ed.) CP 2011. LNCS, vol. 6876, pp. 240–255. Springer, Heidelberg (2011)
23. Zampelli, S., Vergados, Y., Van Schaeren, R., Dulleart, W., Birger, R.: The berth allocation and quay crane assignment problem using a CP approach Technical Report at Universiteit Antwerpen (2013), `http://www.ua.ac.be/main.aspx?c=wout.dullaert&n=9300&ct=005748`