Robust Solutions to Schedule Container Vessels in Terminal Ports

Abstract

Container terminals are open systems that carry out a large number of planning and scheduling problems. Most of these problems must be solved to obtain optimal/suboptimal solutions according to different objective functions (minimized makespam, waiting times, maximizing profit, etc.). Nevertheless, due to the dynamism of these problems and uncertain information, container operators desire to obtain robust solutions that are able to absorb incidences due to delays, changes in estimations, machine breakdown, etc. In this paper, we consider the problem of scheduling a number of incoming vessels by assigning a berthing position, a berthing time, and a number of Quay Cranes to each vessel. This problem is known as the Berth Allocation Problem and the Quay Crane Assignment Problem. We propose a GRASP-based metaheuristic for solving these scheduling problems by finding optimized and robust solutions (multiobjective). Thus, our aim is to minimize the total service time elapsed to serve all the vessels but maximizing the temporal distance between consecutive vessels. The different cases studied verify the optimality and robustness of the solutions obtained.

Introduction

Nowadays, many planning and scheduling techniques have recently seen important advances thanks to the application of heuristics and metaheuristics techniques. Most real-world problems can be cast as highly coupled planning and scheduling problems, where resources must be allocated so as to optimize overall performance objectives. Therefore, solving these problems requires an adequate mixture of planning, scheduling and resource allocation to competing goal activities over time in the presence of complex state-dependent constraints. Most of these problems are very complex and sometimes they must be optimized according to opposite goals (multiobjective criteria). Thus, they must be simplified with several assumptions in order to be appropriately modeled. One alternative option for studying these real-world systems is to use simulation. This alternative provides the opportunity to represent each part of the system easily and to observe how they work with each other.

The container terminal is one of these systems, where there are several machines (quay cranes, yard cranes, trucks, etc.) with different characteristics to ensure the fast loading and unloading of the vessels.

Container terminals generally serve as a transshipment zone between ships and land vehicles (trains or trucks). In (Henesey 2006), Henesey shows how this transshipment market is growing fast. Between 1990 and 2008, container traffic has grown from 28.7 million to 152.0 million, an increase by about 430%. This corresponds to an average annual compound growth of 9.5%. In the same period, container throughput went from 88 million to 530 million of containers, an increase of 500%, equivalent to an average annual compound growth of 10.5%. The surge of both container traffic and throughput is linked with the growth of international trade in addition to the adoption of containerization as privileged vector for maritime shipping and inland transportation (rep 2010). However, the Global Economic Crisis of 2008 has had a negative impact over the container traffic (Mohi-Eldin and Mohamed 2010). This situation increases the competition among the container terminals and forces the container terminal to offer their better services to shipping lines.

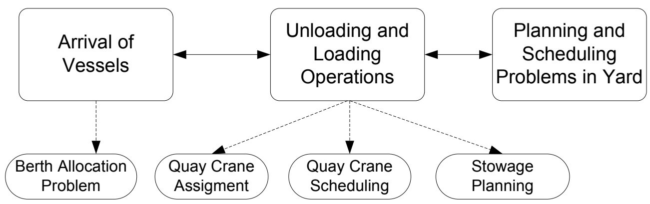


Figure 1: Planning and scheduling problems in Container Terminals.

Container terminals are open systems with three distinguishable areas: the berth area, where vessels are berthed for service; the storage yard, where containers are temporarily stored to be exported or imported; and the terminal receipt and delivery gate area, which connects the container terminal to the hinterland. Each one presents different planning and scheduling problems to be optimized (Lim 1998a; Imai et al. 2008). For example, berth allocation, quay crane assignment, stowage planning, and quay crane scheduling must be managed in the berthing area; the container stack-

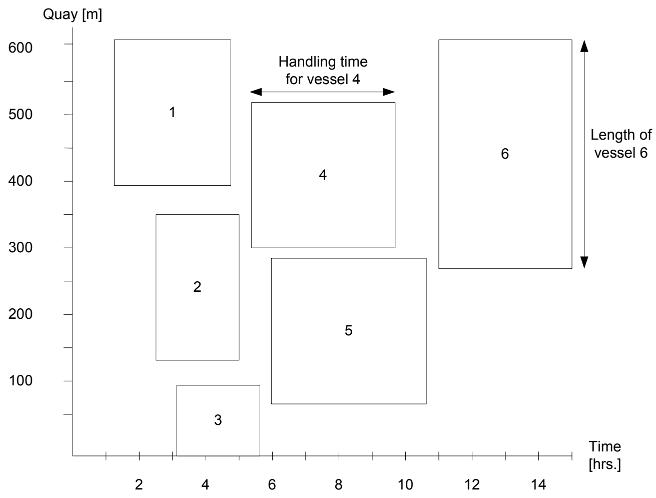


Figure 2: A berth plan with 6 vessels.

ing problem, yard crane scheduling, and horizontal transport operations must be carried out in the yard area; and the hinterland operations must be solved in the landside area. Figure 1 shows the main planning and scheduling problems that must be managed in a container terminal.

Managers at container terminals are confronted with two interrelated decisions: *where* and *when* the vessels should moor. First, they have to take into account physical restrictions such as length or draft, and they also have to take into account priorities and other aspects to minimize both port and user costs, which are usually opposites. Generally, this process is solved manually. It is usually solved by means of a policy to serve the first vessel that arrives (FCFS). Figure 2 shows an example of the graphical space-time representation of a berth plan with 7 vessels. Each rectangle represents a vessel with its handling time and length. For instance, vessel 4 must moor after vessels 1 and 2 depart.

Research work carried out on optimization of port terminal operations suggests that integrated planning of related port operations significantly enhances the terminal efficiency with a more effective use of limited resources of the port (Umang, Bierlaire, and Vacca 2011).

We focus our attention on two planning and scheduling problems, the Berth Allocation Problem (BAP) and the Quay Crane Assignment Problem (QCAP). The former is a well-known combinatorial optimization problem (Lim 1998b), which consists of assigning incoming vessels to berthing positions. The latter deals with assigning a determined number of QCs to each vessel that is waiting at the roadstead such that all required movements of containers can be fulfilled. Once a vessel arrives at the port, it waits at the roadstead until it has permission to moor at the quay. The quay is a platform protruding into the water to facilitate the loading and unloading of cargo. The locations where mooring can take place are called berths. These are equipped with giant cranes, known as Quay Cranes (QC), used to load and unload containers which are transferred to and from the yard by a fleet of vehicles. In a transshipment terminal the yard allows temporary storage before containers are transferred to another ship or to another mode (e.g., rail or road).

Another problem arising in these systems is their own dynamism and lack of information. Failures or breakdowns of quay cranes, straddle carriers, rubber tyred gantry cranes, etc. as well as operators productivity have an impact on the service time of the vessels. That is the reason to include robustness (Verfaillie and Jussien 2005) into the planning and scheduling tasks in order to offer solutions that may remain valid after minor changes.

The remainder of this paper is organized as follows: the next section presents a review of the literature about the BAP and QCAP and different techniques to manage them. Afterwards, the model we consider in this paper is defined. Then, the mathematical model and the developed metaheuristic technique are explained. And finally, the computational results of the experiments as well as the conclusions of this paper are shown.

Literature review

One of the early works that appeared in the literature developed a heuristic algorithm by considering a First-Come-First-Served (FCFS) rule (Lai and Shih 1992). However, the idea that for high port throughput, optimal vessel-to-berth assignments should be found without considering the FCFS bases was introduced in (Imai, Nagaiwa, and Tat 1997). Nevertheless, this approach may result in some vessels' dissatisfaction regarding the order of service.

In (Bierwirth and Meisel 2010), the authors give a comprehensive survey of berth allocation and quay crane assignment formulations from the literature. Some authors outline approaches more or less informally while others provide precise optimization models. More than 40 formulations that are distributed among discrete problems, continuous problems, and hybrid problems are presented. Furthermore, Bierwirth and Meisel (Bierwirth and Meisel 2010) developed a classification scheme to show similarities and differences in the existing models for berth allocation. They classify the BAP according to four attributes. The spatial attribute concerns the berth layout and water depth restrictions. The temporal attribute describes the temporal constraints for the service process of vessels. The handling time attribute determines the way vessel handling times are considered in the problem. The fourth attribute defines the performance measure to reflect different service quality criteria. The most important ones are minimizing the waiting time and the handling time of a vessel. Both measures aims at providing a competitive service to vessel operators. If both objectives are pursued (i.e. wait and hand are set), the port stay time of vessels is minimized. Other measures are focused on minimizing the completion times of vessels among others. Thus, by using the above classification scheme, a certain type of BAP is described by a selection of values for each one of the attributes.

An approach based on multi-objective optimization problem using evolutionary algorithms ((Cheong, Tan, and Liu 2009)) is followed to minimize the makespan of the port, total waiting time of the ships, and degree of deviation from a predetermined service priority schedule. But, this optimization is carried out by considering fixed handling times of the ships. In (Lee and Chen 2009), a neighborhood-search based heuristic is developed to solve this problem where the quay is continuous as well as ship shifting is allowed. These approaches provide optimized solutions to BAP. However, they do not consider assignment of Quay Cranes which is needed to assess the actual required mooring time.

The Quay Crane Assignment Problem (QCAP) has not received much attention in academic research as an independent problem (Bierwirth and Meisel 2010). This problem has been considered in some berthing allocation models.

There are a few studies which involve discrete BAP and QCAP. In (Giallombardo et al. 2010), Giallombardo et al. present the integration of both problems through two mixed integer programming formulations including a tabu search method which is an adaption of the method presented in (Cordeau et al. 2005), however they minimize the yard-related housekeeping costs generated by the flows of containers exchanged between vessels. Liang et al. (Liang, Guo, and Yang 2011) also develop a genetic algorithm with a multiobjective function, total service time and movements of the Quay Cranes. Imai et al. (Imai et al. 2008) present a mathematical formulation and they solve it by means of a Genetic Algorithm in two phases due to the fact of the complexity of this formulation. Regarding handling time, a minimum number of Quay Cranes for each vessel is required to begin loading and unloading tasks. The integration of BAP considering the quay as a continuous line and QCAP was introduced in (Park and Kim 2003), which proposes a non-linear integer programming model.

Our approach studies the integration of these two problems (continuous BAP and QCAP) through a metaheuristic called Greedy Randomized Adaptive Search Procedures (GRASP) (Feo and Resende 1995), taking into account the requirements of container operators of MSC (Mediterranean Shipping Company S.A.). This metaheuristic is able to find optimized scheduling solutions within an acceptable computational time in an efficient way.

Problem description

In this section we present the notation of the main parameters that will be used in both the developed mathematical model and the proposed metaheuristic technique (Figure 3).

The objective in BAP+QCAP is to obtain an optimal distribution of the docks and cranes for vessels waiting to berth. Thus, this problem could be considered as a special kind of machine scheduling problem, with specific constraints (length and depth of vessels, ensuring a correct order for vessels that exchange containers, ensuring departure times, etc.) and optimization criteria (priorities, minimization of waiting and staying times of vessels, satisfaction with the order of berthing, minimizing of crane moves, degree of deviation from a pre-determined service priority, etc.).

Our model is classified according to the classification given by (Bierwirth and Meisel 2010) as:

- *Spatial attribute*: we assume that the quay is a continuous line (*cont*), so there is no partitioning of the quay and the vessel can berth at arbitrary positions within the boundaries of the quay. It must be taken into account that for a continuous layout, berth planning is more complicated than for a discrete layout, but it utilizes quay spaces better (Bierwirth and Meisel 2010).

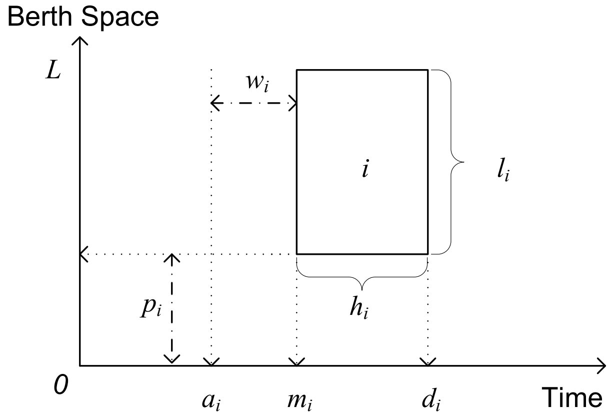


Figure 3: Representation of a vessel according its position and times.

- *Temporal attribute*: we assume dynamic problems (*dyn*) where arrival times restrict the earliest berthing times. Since fixed arrival times are given for the vessels, vessels cannot berth before their expected arrival time.
- *Handling time attribute*: we assume that the handling time of a vessel depends on the assignment of QCs (*QCAP*).
- *Performance measure*: Our objective is to minimize the sum of the waiting time (*wait*) of all the scheduled vessels to be served.

Let V be the set of incoming vessels. Following, we introduce the notation used for each vessel $i \in V$. The variables that represent the problem data are:

- a_i : Arrival time of the vessel i at port.
- c_i : Number of required movements to load and unload containers of i.
- l_i : Vessel length.
- pr_i : Vessel priority.
- QC : Available QCs in the container terminal.
- L : Total length of the berth in the container terminal.

And the decision variables:

- m_i : Moored time of i. Through this variable, waiting time is calculated as:

$$w_i = m_i - a_i \tag{1}$$

- p_i : Berthing position where i will moor.
- q_i : Number of assigned QCs to i.
- u_{ik} : Indicates whether QC k works on the vessel i (1) or not (0).
- h_i : Loading and unloading time at quay (handling time). This time depends on q_i and c_i.

- d_i : Departure time of vessel i:

$$d_i = m_i + h_i \quad (2)$$

Berthing position (p_i) will be determined according to the length of the vessels (and their safety distances) which have been already planned. In addition, this position will be as close to the ends of the quay as possible. Thereby, there will be more continuous length available for the remaining vessels.

Basically, our objective is to allocate all vessels according to several constraints with the objective of minimizing the total service time. To this end, and following shipowner criteria, we estimate a priority of each vessel according to its length and the number of movements (loading and unloading operations), in order to avoid the vessels' dissatisfaction mentioned above.

$$pr_i = \frac{\alpha \times l_i + \beta \times c_i}{l_i + c_i} \quad (3)$$

where α and β are the needed factors to distribute the values of l_i and c_i in order to range the priority in $[0, 1]$. The values of α and β can be given by the operators (Container Terminal). For instance, values of α and β could be:

$$\alpha = \begin{cases} 0 & 0 < l_i < 50 \\ 0.5 & 50 \leq l_i < 150 \\ 1 & l_i \geq 150 \end{cases} \qquad \beta = \begin{cases} 0 & 0 < c_i < 50 \\ 0.5 & 50 \leq c_i < 250 \\ 1 & c_i \geq 250 \end{cases}$$

Thus, a vessel, with length $l_i = 80$ and the number of required movements is $c_i = 150$, has a priority:

$$pr_i = \frac{0.5 \times 80 + 0.5 \times 150}{80 + 150} = 0.5$$

Nevertheless, the priority of each vessel can be reassigned according to any other shipowner's criteria.

As we have pointed out, we deal with continuous and dynamic BAP and the time is discretized into integer units $(1, 2, \ldots, T)$. Moreover, we consider the following assumptions:

- Number of QCs assigned to a vessel do not vary along the moored time. Moreover, all QCs carry out the same number of movements by unit time (`movsQC`, given by the container terminal).
- All the information related to the waiting vessels is known in advance.
- Every vessel has a draft lower or equal than the quay.
- Mooring and unmooring are no time consuming as well as the movement of QCs.
- Simultaneous berthing is allowed.

Therefore, in order to allocate one vessel at berth, the following constraints must be accomplished:

- Moored time must be at least the same that its arrival time:

$$m_i \geq a_i$$

- There is enough contiguous space at berth to moor the vessel (l_i).
- There is a safety distance (`secLength`) between two moored ships: we assume 5% of the length of the vessels (the maximum of these two contiguous vessels).
- There must be at least one QC to assign to each vessel. The maximum number of assigned QCs by vessel depends on its length, since a safety distance is required between two contiguous QCs (`secQC`), and the maximum number of QCs that the container terminal allows per vessel (`maxQC`). Both parameters are given by the container terminal. Thus, the handling time of i is given by:

$$\frac{c_i}{q_i \times \texttt{movsQC}} \quad (4)$$

- Once a QC starts a task in a vessel, it must complete it without any pause or shift (non-preemptive tasks).

Algorithm 1: Obtaining secLength and maxQC for each vessel.

Data: V: set of Vessels
Result: `sd` security distances; $\texttt{QC}^+$ maximum number of QCs per vessel

```
foreach v ∈ V do
    QC_v^+ := max(l_v / secQC, 1);
    QC_v^+ := min(QC_v^+, maxQC);
    foreach w ∈ W do
        if l_v > l_w then
            sd_vw := ⌈l_v × secLength⌉;
        else
            sd_vw := ⌈l_w × secLength)⌉;
        end
    end
end
```

Following the previous constraints, the maximum number of QCs of each vessel $v \in V$ (QC_v^+) as well as the security distance between each pair of vessels ($sd_{vw}, v, w \in V$) are calculated in advance by Algorithm 1.

The goal of the BAP is to minimize the total weighted waiting time of vessels:

$$T_w = \sum_i w_i^\gamma \times pr_i \quad (5)$$

when QCAP is also considered, this function becomes minimizing the total service time:

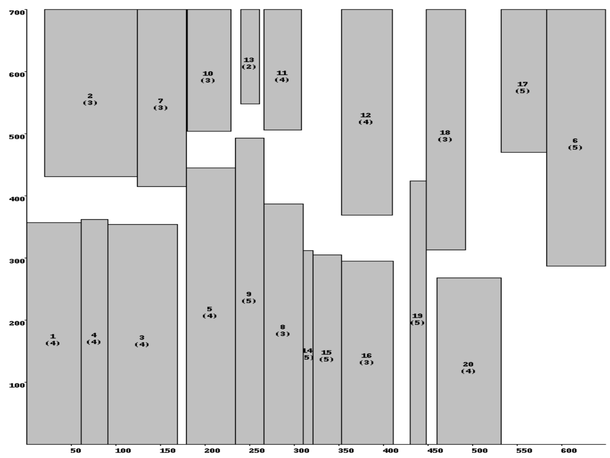
$$T_s = \sum_i (w_i^\gamma + h_i) \times pr_i \quad (6)$$

where $\gamma : (\gamma \geq 1)$ is an adjustment factor to prevent that lower priority vessels are systematically delayed. Note that this objective function is different to the tardiness concept in scheduling. The weighted optimization of tardiness of vessels would be:

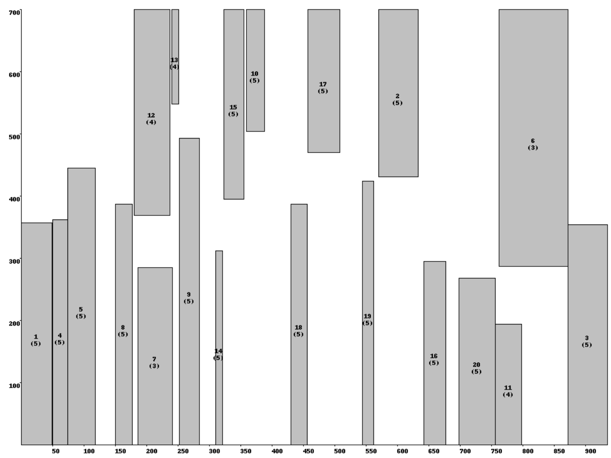
$$T_t = \sum_i (d_i - dueTime(i)) \times pr_i \quad (7)$$

so that the departure time (d_i) of vessels with respect to their due times ($dueTime(i)$) is optimized.

When robustness is taken into account, not only do operators require to minimize the service time (Figure 4(a), but



(a) Minimizing service time ($T_s = 17423$; $R = 128.69$)



(b) Maximizing robustness ($T_s = 32996$; $R = 1099.49$)

Figure 4: Possible berth plans

they need to maximize the temporal distance among the following vessels (Figure 4(b)) to be employed in case a failure or delay occurs. This objective function is calculated as shown in Equation 8.

$$R = \sum_{i \in V} \sum_{j \in V} \lambda_{ij} \frac{m_j - d_i}{h_i} \tag{8}$$

where λ_{ij} means that vessel j has moored after vessel i and j would be delayed if i would suffer any delay during its handling time. This function depends on two measures. The first measure is the free time between every two vessels i and j ($m_i < m_j$). And the second measure is the handling time of the vessel i. This is because it is more likely that the breakdowns arise when there are many containers for loading and unloading. Thus, it seeks to achieve solutions that the greater their handling time of vessel i, the greater the following time until de mooring time of the vessel j.

Equation 9 relates the total weighted service time function to the robustness function, where service time is minimized and robustness is maximized. There are also two factors (α and β) each one associated to a different objective function. Factors α and β must be adjusted according to the required relevance.

$$F = \alpha \times \frac{T_s}{|V|} - \beta * \log(R) \tag{9}$$

Mathematical Formulation

In this section, the mathematical formulation for BAP+QCAP is presented. This formulation is based on the model presented in (Kim and Moon 2003), but adding the needed constraints to take into consideration QCs. Thereby, the handling time depends on the number of QCs and these QCs cannot pass each other when are relocated.

Thus, BAP+QCAP are then modeled as follows:

$$\min \sum_{i \in V} (w_i^{\gamma} + h_i) \times pr_i \tag{10}$$

subject to:

$$m_i \geq a_i \quad \forall i \in V \tag{11}$$

$$w_i = m_i - a_i \quad \forall i \in V \tag{12}$$

$$p_i + l_i \leq L \quad \forall i \in V \tag{13}$$

$$q_i = \sum_{k \in QC} u_{ik} \quad \forall i \in V \tag{14}$$

$$1 \leq q_i \leq QC_i^{+} \quad \forall i \in V \tag{15}$$

$$s_i \geq e_i \quad \forall i \in V \tag{16}$$

$$q_i = e_i - s_i + 1 \quad \forall i \in V \tag{17}$$

$$\sum_{k \in QC} t_{ik} \times \texttt{movsQC} \geq c_i \quad \forall i \in V \tag{18}$$

$$h_i = \max_{k \in QC} t_{ik} \quad \forall i \in V \tag{19}$$

$$t_{ik} - u_{ik} \times M \leq 0 \quad \forall i \in V, \forall k \in QC \tag{20}$$

$$h_i - M \times (1 - u_{ik}) - t_{ik} \leq 0 \quad \forall i \in V, \forall k \in QC \tag{21}$$

$$u_{ik} + u_{jk} + z_{ij}^{x} \leq 2 \quad \forall i, j \in V, \forall k \in QC \tag{22}$$

$$M \times (1 - u_{ik}) + (e_i - k) \geq 0 \quad \forall i \in V, \forall k \in QC \tag{23}$$

$$M \times (1 - u_{ik}) + (k - s_i) \geq 0 \quad \forall i \in V, \forall k \in QC \tag{24}$$

$$p_i + l_i \leq p_j - sd_{ij} + M \times (1 - z_{ij}^{x}) \quad \forall i, j \in V,\ i \neq j \tag{25}$$

$$e_i + 1 \leq s_j + M \times (1 - z_{ij}^{x}) \quad \forall i, j \in V,\ i \neq j \tag{26}$$

$$m_i + h_i \leq m_j + M \times (1 - z_{ij}^{y}) \quad \forall i, j \in V,\ i \neq j \tag{27}$$

$$z_{ij}^{x} + z_{ji}^{x} + z_{ij}^{y} + z_{ji}^{y} \geq 1 \quad \forall i, j \in V,\ i \neq j \tag{28}$$

$$z_{ij}^{x}, z_{ij}^{y}, u_{ik} \quad \text{0/1 integer} \quad \forall i, j \in V,\ i \neq j, \forall k \in QC \tag{29}$$

$$1 \leq s_i, e_i \leq |QC| \quad \forall i \in V \tag{30}$$

where z_{ij}^{x} is a decision variable that indicates if vessel i is located to the left of vessel j on the berth ($z_{ij}^{x} = 1$), $z_{ij}^{y} = 1$ indicates that vessel i is moored before vessel j in time (29). The other decision variable (u_{ik}) indicates that the QC k is working on vessel i when its value is 1. s_i and

e_i are the indexes for the first and last QC used by vessel i respectively. These two variables are employed to ensure that assigned QCs for vessel i will be contiguous.

This MILP model aims to minimize the total service time for all incoming vessels (Equation 10). Constraint 11 ensures that vessels must moor once they arrive at the terminal. 13 guarantees berthing position does not exceed the length quay. Constraints 14, 15, 16 and 17 assign the number of QCs to the vessel i. Constraint 18 establishes the needed handling time to load and unload their containers. Constraint 20 ensures that not assigned QCs do not work. Constraint 21 forces all assigned QCs to work the same number of hours. Constraint 19 assigns the handling time for vessel i. Constraint 22 avoids that one QC is assigned to two different vessels at the same time. Constraints 23 and 24 force the QCs to be assigned contiguously (from s_i up to e_i). Constraint 25 takes into account the security distance between each two vessels. Constraint 26 avoids that one vessel uses a QC which should cross through the others QCs. Constraint 27 avoids that vessel j moors while the previous vessel i is still at the quay. Constraint 28 establishes the relationship between each pair of vessels. Finally, constraint 30 corresponds to the decision variable that indicates the first and last QC used by vessel i.

A Metaheuristic Search

In this paper, a GRASP metaheuristic (Feo and Resende 1995) algorithm is developed for Berth Allocation and Quay Crane Assignment Problem (Algorithm 2). This is a randomly-biased multistart method to obtain optimized solutions of hard combinatorial problems in a very efficient way. The parameter δ ($0 \leq \delta \leq 1$) allows the tuning of search randomization.

This metaheuristic will be compared with the presented mathematical formulation and the FCFS criteria. Generally, a vessel can be allocated at time t when there is no vessel moored in the berth or there are available contiguous quay length and QCs at that time t. However, in FCFS criteria is considered the constraint: $\forall i \in V, m(V_i) \leq m(V_{i+1})$. The FCFS criteria is employed as an upper bound to evaluate our GRASP metaheuristic.

Algorithm 2: Grasp metaheuristic adapted to BAP+QCAP.

Data: δ factor; V_{out} elements;
Result: V_{in}: sequence of moored vessels;

1 $V_{in} \leftarrow \{\}$; $C \leftarrow V_{out}$;
2 **while** $C \neq \varnothing$ **do**
  /* Obtain the cost function of each vessel e */
3 **foreach** $e \in C$ **do**
    /* Different number of QCs are evaluated */
4 **for** $nq \leftarrow (QC_e^+ - 2)$ **to** QC_e^+ **do**
5 MoorVessel(e, nq, V_{in});
6 $V'_{in} \leftarrow V_{in} \cup \{e\}$;
      /* Once e is moored, the service time of the others vessels V_{out} is obtained following FCFS policy */
7 $V_{FCFS} \leftarrow$ FCFS(V_{out}, V'_{in});
8 $S_e := pr_e \times (w_e) + \sum_{v \in V_{FCFS}} (w_v + h_v) \times pr_v$;
      /* Set of moored vessels that stay at the same range of the quay as e */
9 $W \leftarrow \{v \in V_{in} \mid (p_v \leq p_e < p_v + l_v) \vee (p_v < p_e + l_e \leq p_v + l_v)\}$;
      /* Vessel that is just before vessel e */
10 $p := v \in W, d_v \leq m_e \wedge \forall_{w \in W} d_w < m_e \wedge d_w < d_v$;
      /* Vessel that is just after vessel e */
11 $f := v \in W, m_v \geq d_e \wedge \forall_{w \in W} m_w \geq d_e \wedge m_w > m_v$;
12 $R_e := (m_e - d_p) + (m_f - d_e)$;
      /* Cost function related to element e */
13 $f_c(e) := (1 + \alpha S_e)/(1 + \beta R_e)$;
14 **end**
15 **end**
  /* Generate the Restricted Candidate List */
16 $c_{inf} := \min\{f_c(e) \mid e \in C\}$;
17 $c_{sup} := \max\{f_c(e) \mid e \in C\}$;
18 $RCL \leftarrow \{e \in C \mid f_c(e) \leq c_{inf} + \delta \times (c_{sup} - c_{inf})\}$;
  /* Select one candidate randomly */
19 $v := random(RCL)$;
20 MoorVessel(v, V_{in});
  /* Update moored vessels and candidate vessels */
21 $V_{in} \leftarrow V_{in} \cup \{v\}$; $C \leftarrow C - \{v\}$;
22 **end**
23 $V_{in} \leftarrow$ LocalSearch(V_{in});
24 **return** V_{in};

The first phase of this metaheuristic consists of building a solution by adding one element at a time. This algorithm receives as parameters both the δ factor and the set of vessels V_{out} waiting for mooring at the berth. First, all the waiting vessels V_{out} are considered as candidates C. In step 5, each one of the candidate vessels is moored with different number nq of assigned QCs within the current state. In this process the mooring and departure times ($m(V_i)$,$d(V_i)$), and the berthing position ($p(V_i)$) are assigned to V_i. Next, each candidate is evaluated according to the cost function f_c. This cost functions depends on two factors. The first factor is the sum of the service time of the other unmoored vessels if they are moored following the FCFS policy (step 8). The second factor is the time between the vessel e and its previous p and following f vessels (step 9 to 12).

According to the cost function f_c, a restricted candidate list (RCL) is created (step 18). Then, one vessel v is chosen to be moored following the random degree indicated by δ factor and can no longer be modified. Once v is determined, this is added to the set of vessels V_{in} and eliminated from the candidate list C (step 21). This loop is repeated until C is empty, it means that all the vessels have been moored.

Once a solution have been constructed (all vessels in V_{in} are scheduled), the second phase of the metaheuristic looks for solutions in the neighborhood space (step 23). A local search algorithm looks for vessels v that can be assigned more QCs. For each one (v), as its handling time is now shorter, it is likely that the next vessels (V_N) to v could be moored earlier and so, improving the previous solution. To do this, it is checked whether the vessels V_N satisfy all the constraints to moor earlier keeping the same berthing position and number of QCs from that new mooring time until their new departure time. Note that, at this moment, the handling times of the vessels V_N are not modified.

As the GRASP metaheuristic indicates, this search is repeated according to the number of iterations specified by the user. Furthermore, since all the iterations performed by the GRASP metaheuristic are independent among themselves, this process can be parallelized among different threads easily. Finally, the best solution according to Equation 9 is returned as the solution for the given instance of the problem.

Evaluation

In this section, we evaluate the behavior of the algorithms presented in this paper. The experiments were performed on instances given by the port operators. These instances are composed on a different number of vessels (n) with an exponential distribution of arrivals. The berth length is fixed to 700 meters and the number of Quay Cranes is 7 (corresponding to a determined MSC berth line). The number of required movements and length of vessels are randomly generated between 0 and 900 containers, and between 100 and 475 meters, respectively. Moreover, port operators gave us instances with two different inter-arrival distributions: *Spar* means that the arrival among vessels is sparsely distributed meanwhile *Dens* means that the arrival among vessel is densely distributed.

As we mentioned above, our goal is to minimize the total service time elapsed to serve the set of n incoming vessels. They were solved on a computer equipped with a Core i7-2600 3.40Ghz with 8Gb RAM using 4 threads.

We carried out two different evaluations considering $\gamma = 1$ for objective function. The first evaluation compares the performance of our GRASP metaheuristic against the mathematical model presented above using the CPLEX Solver (12.3). CPLEX was set with a timeout of 60 minutes, and the GRASP algorithm performed iterations until 30 seconds with $\alpha \in [0.1, 0.6]$ depending on the number of incoming vessels. Table 1 shows the best solutions given by CPLEX solver and GRASP method for different number of incoming vessels as well as the time when these solutions were found.

As it can be observed, GRASP metaheuristic got the optimal solutions for the instances with a few incoming vessels. For example, in the second instance our GRASP metaheuristic reached the optimal solutions until 14 vessels. However, CPLEX solver did not have enough computation time to provide the optimal solution from 15 vessels in the first instance and from 9 vessels in the second instance. The weighted service time for those instances is the best found solution when the time out is reached (60 minutes). As it can be seen in Table 1, our metaheuristic approach was able to improve the objective function given by CPLEX in a few seconds. As example, the second instance with 20 vessels, our approach reduced the objective function $T_s = 23663$ down to 16837 after 5 seconds.

In Table 1, the GRASP metaheurstic with robustness was also considered. As it was expected the greater the β parameter, the greater are both measures. For instance, in the first instance with 10 vessels when β was set up to 0.2, T_s increased 100 time units, but R was also increased from 18.7 up to 49.08.

Table 1: A comparison between CPLEX solver and the GRASP metaheuristic (*Dens* inter-arrival distributions)

|V|	**No Robustness Management**						**Robustness Management**		
	CPLEX			**GRASP (α=1; β=0)**			**GRASP (α=0.8; β=0.2)**		
	Time	$\mathbf{T_s}$	**R**	**Time**	$\mathbf{T_s}$	**R**	**Time**	$\mathbf{T_s}$	**R**
5	2.20	971	1.18	0.00	971	1.18	9.21	1074	6.97
6	2.20	1033	2.76	0.00	1033	2.14	10.37	1140	11.34
7	2.75	1285	6.84	0.00	1285	7.54	17.29	1509	23.92
8	3.30	1640	5.81	0.00	1640	23.35	20.92	1695	26.78
9	6.05	1804	20.60	0.01	1804	14.28	15.68	2069	44.15
10	6.54	1872	22.95	0.29	1872	18.70	4.21	1972	49.08
11	17.50	2087	65.36	0.06	2087	49.79	29.13	2239	83.52
12	14.11	2585	52.55	1.75	2585	80.91	12.60	2764	105.06
13	156.19	3602	186.53	0.16	3602	108.86	26.04	3758	141.25
14	24.59	3995	149.50	0.33	3995	200.16	15.19	4149	209.32
15	TimeOut	4313	253.16	4.34	4313	210.67	15.07	4824	372.59
20	TimeOut	6558	538.06	18.10	6415	468.36	0.04	7829	606.25

|V|	**No Robustness Management**						**Robustness Management**		
	CPLEX			**GRASP (α=1; β=0)**			**GRASP (α=0.8; β=0.2)**		
	Time	$\mathbf{T_s}$	**R**	**Time**	$\mathbf{T_s}$	**R**	**Time**	$\mathbf{T_s}$	**R**
5	3.28	3663	0.11	0.00	3663	0.11	14.86	3776	5.42
6	3.85	5543	0.18	0.00	5543	0.18	14.86	5919	12.17
7	8.25	6244	1.40	0.60	6442	1.25	3.51	6852	12.15
8	10.83	6904	0.39	6.31	7168	1.28	8.09	7954	22.72
9	31.14	8033	0.55	4.84	8405	0.55	21.85	9108	17.70
10	120.94	8519	1.31	11.21	9009	5.67	18.38	10043	38.52
11	3001.49	9353	0.77	20.23	9754	6.16	28.53	11432	60.11
12	TimeOut	11894	1.47	9.89	11485	11.39	12.18	13246	74.52
13	TimeOut	11762	14.59	18.97	11546	11.05	27.01	13453	69.41
14	TimeOut	12446	23.86	5.71	12007	7.84	9.94	14329	92.14
15	TimeOut	14397	48.22	19.97	12706	22.01	2.92	15356	155.47
20	TimeOut	23663	143.04	5.00	16837	43.77	12.77	22217	338.35

The relationship between these two functions is shown in Figure 5. In this figure the average values for T_s and R are presented according to the parameters α and β ($\beta = 1 - \alpha$). Note that if it is required a low service time, it is not possible to achieve a solution with a high robustness; and viceversa, if it is required a solution with high robustness, the service time of the solutions found should be high.

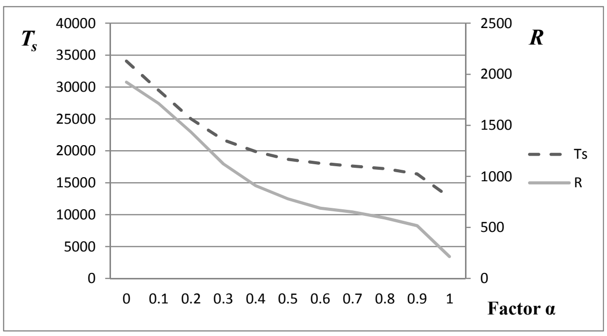
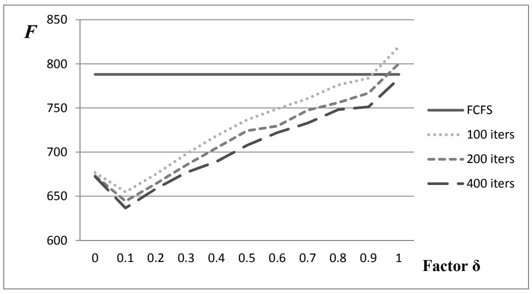
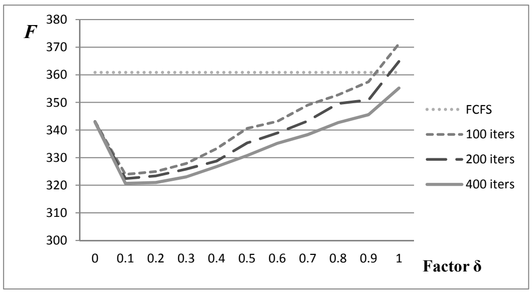


Figure 5: T_s and R evolution when α increases.

One characteristic of the GRASP metaheuristic is observed in Figure 6 and is that the greater the number of it-



(a) *Dens* inter-arrival distribution



(b) *Spar* inter-arrival distribution

Figure 6: Average values of F for 20 vessels.

erations, the better the solutions quality. This tendency is followed with the both *Dens* and *Spar* inter-arrival distributions. For instance, in Figure 6(a) for $\delta = 0.2$ the value of F was 654.8 in *Dens* inter-arrival distributions and was decreased to 636.6 in *Spar* inter-arrival distributions.

Conclusions

Container terminals have to face increasing competitiveness which requires greater efficiency in both quayside and yard. Optimizing the planning and scheduling of your resources and workforce is crucial in operating a competitive terminal to increase throughput. Research work carried out on optimization of port terminal operations suggests the integration of planning and scheduling tasks to enhance the terminal efficiency with a more effective use of resources. Furthermore, due to the dynamism of these tasks it is crucial to include robustness in these tasks to minimize the effect of disruptions in operations and enable fast recovery in real time. We focus our attention on two planning and scheduling problems, the Berth Allocation Problem (BAP) and the Quay Crane Assignment Problem (QCAP). To this end, we propose a GRASP metaheuristic to solve these two problems efficiently. Our GRASP metaheuristic was compared with a mathematical formulation and it was observed that the solutions given are close to the optimal. Furthermore, this metaheuristic has also been adapted to include robustness in the solutions. It means that the solutions given could remain valid although there could be some possible breakdowns in jobs related to the yard storage.

In further works, we will focus our attention on scheduling vessels with fuzzy arrivals and fuzzy handling times.

References

Bierwirth, C., and Meisel, F. 2010. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research* 202:615–627.

Cheong, C.; Tan, K.; and Liu, D. 2009. Solving the berth allocation problem with service priority via multi-objective optimization. In *Computational Intelligence in Scheduling, 2009. CI-Sched '09. IEEE Symposium on*, 95 –102.

Cordeau, J.; Laporte, G.; Legato, P.; and Moccia, L. 2005. Models and tabu search heuristics for the berth-allocation problem. *Transportation science* 39(4):526–538.

Feo, T., and Resende, M. 1995. Greedy randomized adaptive search procedures. *Journal of Global Optimization* 6(2):109–133.

Giallombardo, G.; Moccia, L.; Salani, M.; and Vacca, I. 2010. Modeling and solving the tactical berth allocation problem. *Transportation Research Part B: Methodological* 44(2):232 – 245.

Henesey, L. 2006. Overview of Transshipment Operations and Simulation. In *MedTrade conference, Malta, April*, 6–7.

Imai, A.; Chen, H.; Nishimura, E.; and Papadimitriou, S. 2008. The simultaneous berth and quay crane allocation problem. *Transportation Research Part E: Logistics and Transportation Review* 44(5):900–920.

Imai, A.; Nagaiwa, K.; and Tat, C. 1997. Efficient planning of berth allocation for container terminals in Asia. *Journal of advanced transportation* 31(1):75–94.

Kim, K., and Moon, K. 2003. Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological* 37(6):541–560.

Lai, K. K., and Shih, K. 1992. A study of container berth allocation. *Journal of Advanced Transportation* 26(1):45–60.

Lee, Y., and Chen, C. 2009. An optimization heuristic for the berth scheduling problem. *European Journal of Operational Research* 196(2):500–508.

Liang, C.; Guo, J.; and Yang, Y. 2011. Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning. *Journal of Intelligent Manufacturing* 22:471–479.

Lim, A. 1998a. The berth planning problem. *Operations Research Letters* 22(2-3):105–110.

Lim, A. 1998b. The berth planning problem. *Operations Research Letters* 22(2-3):105–110.

Mohi-Eldin, E., and Mohamed, E. 2010. The impact of the financial crisis on container terminals (a global perspectives on market behavior).

Park, Y., and Kim, K. 2003. A scheduling method for berth and quay cranes. *OR Spectrum* 25(1):1–23.

2010. Global container terminal operators annual review and forecast. *Annual Report*.

Umang, N.; Bierlaire, M.; and Vacca, I. 2011. The berth allocation problem in bulk ports. *Proceedings of the 11th Swiss Transport Research Conference (STRC)*.

Verfaillie, G., and Jussien, N. 2005. Constraint solving in uncertain and dynamic environments: A survey. *Constraints* 10(3):253–281.