A combined Mixed Integer Programming model of seaside operations arising in container ports

Ghazwan Alsoufi, Xinan Yang, Abdellah Salhi
Department of Mathematical Sciences, University of Essex, UK

Abstract

This paper puts forward an integrated optimisation model that combines three distinct problems, namely the Berth Allocation Problem, the Quay Crane Assignment Problem, and the Quay Crane Scheduling problem, which have to be solved to carry out these seaside operations in container ports. Each one of these problems is complex to solve in its own right. However, solving them individually leads almost surely to sub-optimal solutions. Hence the need to solve them in a combined form. The problem is formulated as a mixed-integer programming model with the objective being to minimise the tardiness of vessels. Experimental results show that relatively small instances of the proposed model can be solved exactly using CPLEX.

1 Introduction

Container terminals are important assets in many modern economies. They constitute important means of distributing goods made overseas to domestic markets in other countries. They are expensive to build, and difficult to operate. We describe here some of the main problems which are faced daily by decision makers in container ports.

Container ports consist of a seaside component and a landside component. Seaside operations lead to three problems:

1. the Berth Allocation Problem (BAP),
2. the Quay Crane Assignment Problem (QCAP),
3. the Quay Crane Scheduling Problem (QCSP).

The landside operations include yard planning, yard crane assignment, and container storage planning among other things. In this paper, the seaside problems listed above, will be discussed. Each one is a complex optimization problem in its own right. However, solving these problems individually without consideration of the others may lead to overall suboptimal solutions. For this reason we will investigate the triple integrated problem. An overview of these three problems individually and integrated pairwise can be found in the paper of Bierwirth and Meisel [2].

Operations at container terminals are usually sequenced as follows. The first operation is the berth allocation which allocates a berthing time and a berthing position to every vessel arriving at the port. The main objective of this operation is to minimise the handling time, i.e. load or unload the containers into and from a vessel, depending on the distance between the berthing position and the desired berthing. The desired berthing is the berthing position which has the minimum distance between the berthing position and the pre-allocated yard stowage in the port, where the containers will be stored until they are transferred to other vessels, trains or vehicles which will take them to their next destinations. The decision makers in the container terminal try to find the best position for every vessel to minimize movement cost. Also, they try to find the optimal berthing time for each vessel arriving to the port. The second operation, leading to QCAP, tries to determine the optimum number of quay cranes to allocate to every vessel so that the throughput of the cranes is maximized or, equivalently, their idle time is minimised. Equivalently, the handling time of vessels will be minimized, as a result the quay cranes will finish processing each vessel in optimal time (due time). The decision maker in this type of situation focuses on determining the most suitable number of quay cranes for each vessel. The last operation, leading to QCSP, investigates what is the optimum order in which to carry out tasks in order to minimize the finishing time of processing vessels. When the quay crane performs a task quickly, there is a chance to move to another task on the same vessel or on other vessels. In fact, there are no studies, to our knowledge, that allow quay cranes that have finished their work on a given ship, to move from that ship to another even if the vessel they are leaving is still being processed. More specifically, our model allows the number of quay cranes assigned to a vessel to change during the handling of the vessel.

This paper is organised as follows: Section 2 is a literature review. The problem description will be discussed in Section 3. The proposed mathematical model will be given in Section 4. Section 5 records computational experience on 25 instances. Section 6 is the conclusion and future work.

2 Literature Review

There is, to our knowledge, no previous investigation of the three problems listed above combined into a single model. There is however a lot of work on some individual problems and the pairwise combination of these and others. We will review this literature briefly in the following.

2.1 The Berth Allocation Problem

Imai et al. [9] addressed the problem of determining dynamically berth allocation. They developed a heuristic based on the Lagrangian relaxation of the original problem. Nishimura et al. [22] addressed the dynamic BAP and developed a heuristic method based on the genetic algorithm to solve it. Golias et al. [7] formulated the discrete and dynamic BAP as a multi-objective combinatorial optimization problem. They divided the vessels into groups of vessels each one of which has a different priority from that of other groups. A genetic algorithm based heuristic is then developed to solve it. The heuristic provided a complete set of solutions that enable terminal operators to evaluate various berth scheduling policies and select the schedule that improves operations and customer satisfaction. Cheong et al. [4] solved BAP using multi-objective optimization in order to minimize concurrently the three objectives of makespan, waiting time, and degree of deviation from a predetermined priority schedule. These three objectives represent the interests of both port and ship operators. Unlike most existing approaches using single objective representation, a multi-objective evolutionary algorithm (MOEA) [17, 34] that incorporates the concept of Pareto optimality is proposed for solving the multi-objective BAP. Ting et al. [31] studied the discrete dynamic BAP and suggested a mixed integer programming model with dynamic arrival times. They proposed a particle swarm optimization heuristic to solve it. Lee [13] proposed a neighborhood-search based heuristic to determinate the berthing time and space for each incoming vessel at the continuous berth stretch. In their model, they considered the First-Come-First-Served rule, clearance distance between vessels, and the possibility of vessel shifting. Also, the number of practical considerations was included in this search which is obtained from the port of Kaohsiung. Tong [32] proposed an optimization model for a dynamic berth allocation with discrete layout. The genetic algorithm to determine the berthing time and berthing position at a container terminal was applied. The service priority for each vessel was considered.

2.2 The Quay Crane Scheduling Problem

Kim and Park [11] studied QCSP. They formulated it as a mixed integer programming problem and solved it with Branch-and-Bound (B&B) combined with a heuristic search algorithm, known as the Greedy Randomized Adaptive Search Procedure (GRASP), to overcome the difficulties of B&B alone.

Moccia [20] formulated QCSP as a vehicle routing problem with additional constraints like the precedence relationships between tasks. CPLEX was used to solve small scale instances and Branch-and-Cut to solve large scale instances. Sammarra et al. [29] proposed a Tabu Search heuristic for QCSP in order to minimize the completion time of loading and unloading containers into and from vessels. They considered precedence and non-simultaneity between tasks. Also, they observed that QCSP can be decomposed into a routing problem and a scheduling problem. Lee et al. [12] presented a mixed integer programming model and proved that QCSP is NP-complete. They proposed a genetic algorithm to obtain near optimal solutions. Bierwirth and Meisel [1] noticed that the constraints for interference among the quay cranes still need correction for this reason they revised the previous mathematical models. Chung and Choy [5] proposed a modified genetic algorithm to solve QCSP. In this paper, they used Kim and Park's model. The results were compared with the existing algorithms. The comparison demonstrated that the proposed algorithm is as good as many existing algorithms. Kaveshgar et al. [10] used a genetic algorithm for solving QCSP. Their algorithm improves the efficiency of genetic algorithm search by using an initial solution based on the S-LOAD rule and by reducing the number of genes in the chromosomes to minimize search time. Nguyen et al. [21] suggested two representations based on the genetic algorithm and the genetic programming for QCSP. The genetic algorithm uses permutation to decide the priority of the tasks, whereas the genetic programming procedure relies on a priority function to calculate the priority of tasks.

2.3 The Berth Allocation and Quay Crane Assignment Problems (BACAP)

Meisel and Bierwirth [18] integrated BAP, and QCAP into BACAP. The proposed problem is formulated taking into account some of the real issues faced by the decision maker in the port such as the decrease of marginal productivity of quay cranes assignment to a vessel and the increase in handling time if vessels are not berthed at their desired position at container terminal. In additional to the mathematical model, the authors submitted two meta-heuristic approaches squeaky wheel optimization (SWO), and tabu search (TS). Cheong et al [3] described the benefit of multi-objective optimization on BACAP. The BACAP involves the simultaneous optimization of two highly-coupled container terminal operations. Optimization results show that the multi-objective optimization approach offers the port manager flexibility in selecting a desirable solution for implementation. Liang et al [15] introduced the dynamic quay crane assignment (QCDA) in berth allocation planning problem (BAP) and formulate a multi-objective mathematical model considering each berth for container ship with and the number of Quay Cranes Move. In order to solve this QCDA in BAP problem they proposed a multi-objective hybrid Genetic Algorithm approach with a priority-based encoding method. Yang et al. [33] proposed an approach to solve BACAP as a simultaneous solution of BAP, and QCAP. The mathematical model was formulated by integrating the constraints for BAP of Guan [8] and the constraints for QCAP of Legato [14]. The objective function for this model is the two objective functions for the two previous models of BAP and QCAP. Evolutionary algorithm with nested loops was developed to find the solution for the combined problem.

2.4 The Quay Crane Assignment and Scheduling Problems (QCASP)

Daganzo [6], and Peterkofsky[23] first discussed the combine of QCAP, and QCSP by assignment the number of quay cranes for a set of vessels. In both of these studies, the authors assumed that there is just one task in each bay and did not consider the interference between quay ceanes. Also, they noticed that BAP should be integrated with QCAP, and QCSP. Tavakkoli et al. [30] studied QCASP. In this paper, they formulated a mix integer program to determinate the optimal number of quay cranes for every vessel that will arrive at the terminal and at the same time finding the optimal sequence of the tasks which need to handle on the vessel. Also, a genetic algorithm is proposed to solve the large scale for this type of problem.

2.5 The Berth Allocation, Quay Crane Assignment, and Quay Crane Scheduling Problems (BACASP)

Liu et al. [16] studied the seaside operations and introduced a mixed integer linear programming in order to minimize the maximum relative tardiness of vessel departures. The plausible idea which presented by Liu was that instead of assuming a function relationship between the processing time of a vessel and the number of quay cranes assigned to it, the authors introduced a series of parameters p_{vj} which means the handling time of vessel j when v quay cranes were assigned to it. However, the integration model proposed need further improvement since in the Berth-level model, the berthing times are revised in this model whereas the berthing positions are taken from the tentative berth plan. Meisel et al. [19] proposed a three-phase integration framework using preprocessing and feedback mechanisms. Phase I estimate productivity rates of quay cranes from the vessel stowage plan. In the phase II, by using Meisel's model [18] the berth allocation problem and quay crane assignment problem will be solved depending on the productivity rates of quay cranes. Phase III will schedule the tasks for each vessel. To adjust the solution for BAP, QCAP, and QCSP a feedback loop will be needed. Based on this framework, seaside planning problems for 40 vessels are solvable on a high level of detail within practical computation time.

3 Problem description

The berth allocation problem attempts to find the best time for berthing and the best position for mooring the vessels that arrive at the container terminal. This is an important problem that attracted a lot of attention. After berth allocation is solved, QCAP is the follow up challenge. The number of quay cranes that will be assigned to every vessel is very important. If the number of quay cranes assigned to a vessel is low, the vessel will spend more time in port. This tardiness has a knock on affect on the operators of this vessel and at the same time on the container terminal, because other vessels will wait to get dealt with. The opposite case is when the number of quay cranes assigned to a vessel is more than what is needed. This leads to high handling costs for this vessel. The other problem which follows immediately is that of finding the sequence in which tasks on each vessel are carried out. Choosing the best sequence to perform all the tasks on each vessel is a very important operation which helps minimize the finishing time of every vessel.

All these problems have been solved either individually or by integrating them pairwise. The results obtained for these problems might be optimal. However, the decision maker cannot separate these problems; individual optimal solutions do not guarantee overall optimality. The diagram below shows the relationship between the berth allocation, quay crane assignment, and quay crane scheduling problems. There are some conditions to be considered when solving these three problems simultaneously. The outputs of the berth allocation are the berthing time T_v and the berthing position P_v. These outputs of berth allocation will be used as input to the quay cranes assignment problem to determine the number of quay cranes for each vessel. The starting time of the q^{th} quay crane on the vessel v S_{qv} should be greater than or equal to the berthing time T_v. The starting time of quay crane S_{qv} should be greater than or equal to that of the ready crane r_q which means the earliest available time of the quay crane. The starting time of quay crane S_{qv_j} should be greater than or equal the completion time of the quay crane C_{qv_i} before moving to other vessels. For this reason the quay crane assignment depends on the output of quay crane scheduling and berth allocation. Finally, after determining the number of quay cranes for each vessel, the quay crane scheduling problem will arise to choose the best sequence of the tasks that will be performed by quay cranes which are assigned to it. Also the time to process the vessel F_{vj} should be greater than or equal to the finishing time F_{vi}. The berthing position which will be allocated to the new arriving vessel should be empty. As a result, the berth plan depends on the output of quay crane scheduling.

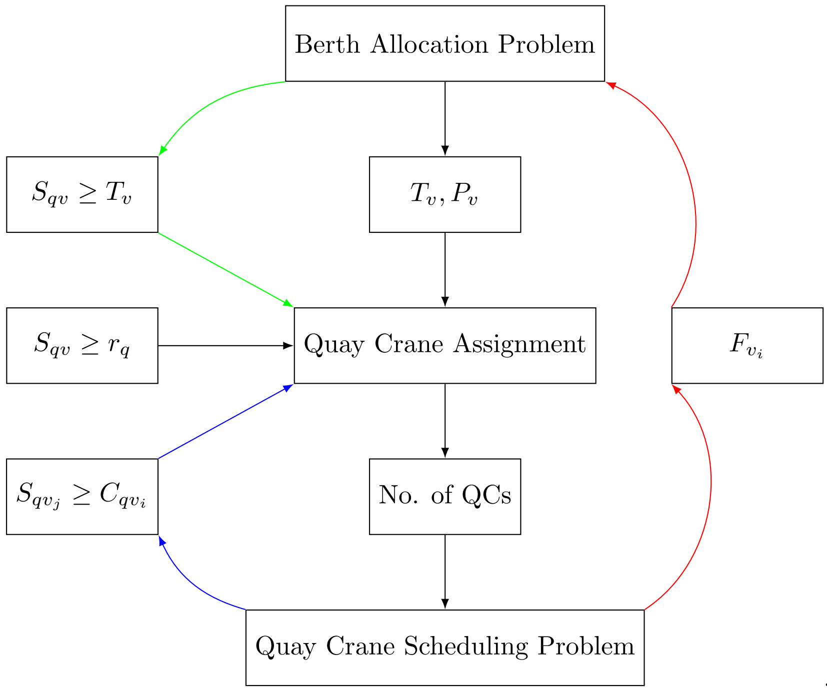

Figure 1: Illustration of the relationship among the three problems

4 Mathematical formulation

This section describes a mixed integer programming model for the continuous and dynamic berth allocation, quay cranes assignment, and quay crane scheduling problems which have been traditionally solved as individual problems. The main aim of this model is to solve these three problems as a one aggregated problem. The suggested mathematical model tries to find the best location and the optimal time for berthing for each vessel that arrives at the container terminal. When the vessel is moored at its preferred position, the distance to transfer the containers from the vessel to the stowage will be minimized and the port will save time to process other vessels and the cost that might be paid to the operator of the vessel if this vessel departs the terminal after the expected time of its departure (due time). Another benefit of this model is to assign enough quay cranes to each vessel. The handling time of the vessel will be longer if an insufficient number of quay cranes is assigned. There are two reasons that prevent us from assigning too many quay cranes to vessel. The first is the cost of building quay cranes which is very expensive; the second is the quay crane constraints i.e. there is a limited number of quay cranes at any time. This model also finds the optimal sequence to perform every task on the vessel; this is the so called quay crane scheduling problem.

4.1 Assumptions

1. Each vessel is divided longitudinally into bays; all bays have the same length. Thus, the length of vessels is in number of bays.
2. The safety distance between each pair of adjacent quay cranes depends on the width of a bay.
3. Each segment of the continuous wharf can handle one vessel at a time.
4. Once a quay crane starts processing a task, it can leave only after it has finished the task.
5. Any vessel can be processed in any space of the wharf depending on the arrival time and the available terminal.

6. Quay cranes are on the same rail and thus they cannot cross over each other.
7. Some tasks must be performed before others and there are tasks that cannot be performed simultaneously.

4.2 Indices

Q Number of quay cranes $(q, q_i, q_j = 1, 2, ...Q)$.
V Number of vessels $(v, v_i, v_j = 1, 2, ..., V)$.
B_v Number of tasks on vessel v $(b, b_i, b_j = 1, 2, ..., B_v)$.

4.3 Parameters

p_b^v Time required to perform task b on vessel v.
l_b^v Location of task b on vessel v expressed by the ship bay number on vessel v.
r_q Earliest available time of the q-th quay crane.
l_0^q Initial location of quay crane q which is relative to the ship-bay number.
l_T^q Final location of quay crane q which is relative to the ship-bay number.
$t_{b_i b_j}^{qv}$ Travel time of the q^{th} quay crane from the location $(l_{b_i}^v)$ of task b_i to the location $(l_{b_j}^v)$ of task b_j on the same vessel v. $t_{b_0 b_j}^{qv}$ Represents the travel time from the initial position (l_q^v) of the q^{th} quay crane to the location $(l_{b_j}^v)$ of the task b_j on vessel v. In addition, $t_{b_j b_{B_v+1}}^{qv}$ represents the travel time from location $(l_{b_j}^v)$of task b_j to the final destination (l_q^v) of q^{th} quay crane on vessel v.
a_v Estimated arrival time for vessel v.
d_v Requested departure time for vessel v.
$\hat{P}_v$ Preferred berth position of vessel v. It is determined by the position of yard storage areas allocated to vessel v. $\hat{P}_v$ reflects that the berth position has the shortest distance to the allocated yard storage areas for vessel v.
U_v Distance cost for vessel v. If vessel v moors at $\hat{P}_v$, the transportation cost is the lowest based on the distance cost due to the vessel mooring at a place with deviation in distance.
L_v Length of the vessel v.
W Length of the wharf.
H_q Variable cost of using the q^{th} quay crane.
W_v Tardiness cost of vessel v per-unit time.
R_v Earliness incoming of vessel v per-unit time.
Ψ Set of pairs of tasks that cannot be performed simultaneously. When tasks b_i and b_j cannot be performed simultaneously, then $((b_i, b_j) \in \{\Psi\})$.
Φ Set of ordered pairs of tasks for which there is a precedence relationship. When task b_i must precede task b_j, then we have $((b_i, b_j) \in \{\Phi\})$.
M Arbitrary large positive number.

4.4 Binary decision variables

$$X_{b_i b_j}^{qv} = \begin{cases} 1 & \text{if the } q^{th} \text{ quay crane performed task } b_j \text{ immediately after performing task } b_i \text{ on vessel } v. \\ 0 & \text{otherwise} \end{cases}$$

Tasks b_0 and b_{B_v+1} are considered as the dummy initial and final states of each quay crane, respectively. Thus, when task b_j is the first task of the q-th quay crane then $X_{b_0 b_j}^{qv} = 1$. In addition, when task b_j is the last task of the q^{th} quay crane then $X_{b_j b_{B_v+1}}^{qv} = 1$.

$$Z_{b_ib_j}^{v} = \begin{cases} 1 & \text{if task } b_j \text{ starts later than the finish of task } b_i \text{ on vessel } v. \\ 0 & \text{otherwise.} \end{cases}$$

$$Y_{v_iv_j}^{q} = \begin{cases} 1 & \text{if the } q^{th} \text{ quay crane is assigned to vessel } v_j \text{ right after finishing its tasks on vessel } v_i. \\ 0 & \text{otherwise.} \end{cases}$$

$$\delta_{v_iv_j} = \begin{cases} 1 & \text{if the processing of vessel } v_j \text{ starts later than the finish time of vessel } v_i. \\ 0 & \text{otherwise.} \end{cases}$$

$$\sigma_{v_iv_j} = \begin{cases} 1 & \text{if the vessel } v_j \text{ is located below the vessel } v_i \text{ in the berth (wharf).} \\ 0 & \text{otherwise.} \end{cases}$$

$$\alpha_{b_ib_j}^{v_iv_j} = \begin{cases} 1 & \text{if the task } b_j \text{ on vessel } v_j \text{ is located below the task } b_i \text{ on vessel } v_i. \\ 0 & \text{otherwise.} \end{cases}$$

$$\beta_{b_ib_j}^{v_iv_j} = \begin{cases} 1 & \text{if the task } b_j \text{ on vessel } v_j \text{ starts later than the finish of task } b_i \text{ on vessel } v_i. \\ 0 & \text{otherwise.} \end{cases}$$

4.5 Continuous decision variables

T_v	Berthing time of vessel v.
P_v	Berthing position of vessel v.
A_v	Tardiness of vessel v.
E_v	Earliness of vessel v.
C_{qv}	Completion time of q^{th} on vessel v.
F_v	Finishing time of vessel v.
$D_{b_i}^{v}$	Completion time of task b_i on vessel v.
S_{qv}	Starting time of q^{th} on vessel v.

4.6 The mathematical model

$$\min Z = \sum_{q=1}^{Q} H_q C_q + \sum_{v=1}^{V} W_v A_v - \sum_{v=1}^{V} R_v E_v + \sum_{v=1}^{V} U_v |P_v - \hat{P}_v| \tag{1}$$

s.t

$$d_v - F_v = E_v - A_v \qquad \forall v \tag{2}$$

$$F_{v_i} \leq T_{v_j} + M(1 - \delta_{v_iv_j}) \qquad \forall v_i, v_j; v_i \neq v_j \tag{3}$$

$$P_{v_i} + L_{v_i} \leq P_{v_j} + M(1 - \sigma_{v_iv_j}) \qquad \forall v_i, v_j; v_i \neq v_j \tag{4}$$

$$\sigma_{v_iv_j} + \sigma_{v_jv_i} + \delta_{v_iv_j} + \delta_{v_jv_i} \geq 1 \qquad \forall v_i, v_j; v_i \neq v_j \tag{5}$$

$$a_v \leq T_v \qquad \forall v \tag{6}$$

$$P_v + L_v \leq W \qquad \forall v \tag{7}$$

$$\sum_{v_j=1}^{V} Y_{v_0v_j}^{q} = 1 \qquad \forall q \tag{8}$$

$$\sum_{v_i=1}^{V} Y_{v_i(V+1)}^{q} = 1 \qquad \forall q \tag{9}$$

$$\sum_{v_j=1}^{V+1} Y_{vv_j}^{q} - \sum_{v_j=0}^{V} Y_{v_jv}^{q} = 0 \qquad \forall v, q \tag{10}$$

$$\sum_{v_i=0}^{V}\sum_{q=1}^{Q} Y^q_{v_i v} \geq 1 \qquad \forall v \tag{11}$$

$$S_{qv} \geq r_q - M(1 - Y^q_{v_0 v}) \qquad \forall v, q \tag{12}$$

$$S_{qv} \geq T_v - M(1 - \sum_{v_j=1}^{V+1} Y^q_{v v_j}) \qquad \forall v, q \tag{13}$$

$$S_{qv_j} \geq C_{qv_i} - M(1 - Y^q_{v_i v_j}) \qquad \forall v_i, v_j; v_i \neq v_j; q \tag{14}$$

$$\sum_{b_j=1}^{B_v} X^{qv}_{b_0 b_j} = \sum_{v_i=0}^{V} Y^q_{v_i v} \qquad \forall v, q \tag{15}$$

$$\sum_{b_j=1}^{B_v} X^{qv}_{b_j b_{B_v+1}} = \sum_{v_i=0}^{V} Y^q_{v_i v} \qquad \forall v, q \tag{16}$$

$$\sum_{b_j=1}^{B_v+1} X^{qv}_{b b_j} - \sum_{b_j=0}^{B_v} X^{qv}_{b_j b} = 0 \qquad \forall b, v, q \tag{17}$$

$$\sum_{q=1}^{Q}\sum_{b_i=0}^{B_v} X^{qv}_{b_i b} = 1 \qquad \forall b, v \tag{18}$$

$$\sum_{b_i=0}^{B_v}\sum_{b_j=1}^{B_v+1} X^{qv}_{b_i b_j} \leq M \sum_{v_i=0}^{V} Y^q_{v_i v} \qquad \forall v, q \tag{19}$$

$$D^v_{b_i} + t^{qv}_{b_i b_j} + p^v_{b_j} - D^v_{b_j} \leq M(1 - X^{qv}_{b_i b_j}) \qquad \forall b_i, b_j; b_i \neq b_j, v, q \tag{20}$$

$$S_{qv} + t^{qv}_{b_0 b_j} + p^v_{b_j} - D^v_{b_j} \leq M(1 - X^{qv}_{b_0 b_j}) \qquad \forall b_j, v, q \tag{21}$$

$$D^v_{b_j} - C_{qv} \leq M(1 - X^{qv}_{b_j b_{B_v+1}}) \qquad \forall b_j, v, q \tag{22}$$

$$C_{qv} - F_v \leq M(1 - \sum_{v_j=1}^{V+1} Y^q_{v v_j}) \qquad \forall v, q \tag{23}$$

$$D^v_{b_i} + p^v_{b_j} \leq D^v_{b_j} \qquad \forall (b_i, b_j) \in \Phi_v; b_j \neq b_i; \forall v \tag{24}$$

$$D^v_{b_i} - D^v_{b_j} + p^v_{b_j} \leq M(1 - Z^v_{b_i b_j}) \qquad \forall b_i, b_j; b_i \neq b_j; \forall v \tag{25}$$

$$Z^v_{b_i b_j} + Z^v_{b_j b_i} = 1 \qquad \forall (b_i, b_j) \in \Psi_v; b_j \neq b_i; \forall v \tag{26}$$

$$\sum_{\theta=0}^{Q}\sum_{\kappa=0}^{B_v} X^{\theta v}_{\kappa b_j} - \sum_{\theta=0}^{Q}\sum_{\kappa=0}^{B_v} X^{\theta v}_{\kappa b_i} \leq M(Z^v_{b_i b_j} + Z^v_{b_j b_i}) \qquad \forall b_i, b_j; b_i \neq b_j; l_{b_i} < l_{b_j}; \forall v, q \tag{27}$$

$$P_{v_i} + l^v_{b_i} \leq P_{v_j} + l^v_{b_j} + M(1 - \alpha^{v_i v_j}_{b_i b_j}) \qquad \forall b_i, b_j, v_i, v_j; v_j \neq v_i \tag{28}$$

$$D^{v_i}_{b_i} - D^{v_j}_{b_j} + p^{v_j}_{b_j} \leq M(1 - \beta^{v_i v_j}_{b_i b_j}) \qquad \forall b_i, b_j, v_i, v_j; v_j \neq v_i \tag{29}$$

$$\beta^{v_i v_j}_{b_i b_j} + \beta^{v_j v_i}_{b_j b_i} + \alpha^{v_i v_j}_{b_i b_j} \geq \sum_{\kappa=0}^{Bv} X^{q_i v_i}_{\kappa b_j} + \sum_{\kappa=0}^{Bv} X^{q_j v_j}_{\kappa b_i} - 1 \qquad \forall b_i, b_j, v_i, v_j, q_i, q_j; v_j \neq v_i; q_i < q_j \tag{30}$$

$$X^{qv}_{b_i b_j}, Z^v_{b_i b_j}, Y^q_{v_i v_j}, \delta_{v_i v_j}, \sigma_{v_i v_j}, \alpha^{v_i v_j}_{b_i b_j}, \beta^{v_i v_j}_{b_i b_j} \in \{0, 1\} \tag{31}$$

$$C_{qv}, F_v, D^v_{b_j}, P_v, T_v \geq 0 \tag{32}$$

In the objective function (1), the first term $\sum_{q=1}^{Q} H_q C_q$ represents the cost of using quay cranes. The second term $\sum_{v=1}^{V} W_v A_v$ represents the tardiness cost if the departure time of the vessel is greater than its due time. The third term $\sum_{v=1}^{V} R_v E_v$ represents the earliness incoming if the finishing time of vessel less than the due time. The last term $\sum_{v=1}^{V} U_v |P_v - \hat{P}_v|$ represents the cost if the vessel moored at an undesired berthing position. Constraint (2) determines if the vessel has earliness or tardiness depending on the difference between the due time of the vessel and the finishing time of this vessel.

The constraints (3-7) represent the conditions for berth allocation. Wherever, constraint (3) define $\delta_{v_i v_j}$ such that $\delta_{v_i v_j} = 0$ or 1 if the finishing time of vessel i less than or equal the berthing time of vessel j; 0 if the finishing time of vessel i greater than the berthing time of vessel j. The figures below illustrate how the value of $\delta_{v_i v_j}$ is computed:

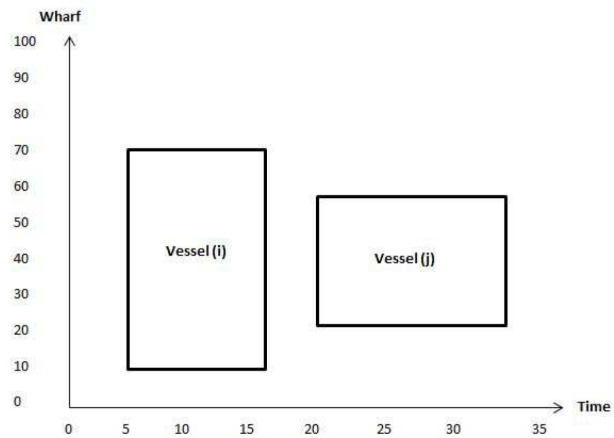

Figure 2: Illustration of no time overlap between two vessels

The value of $\delta_{v_i v_j}$ in the Fig.2 equal 0 or 1 because $F_{v_i} \leq T_{v_j}$. Whereas, the value of $\delta_{v_j v_i}$ in the same figure equal 0 because $F_{v_j} > T_{v_i}$. The value of $\delta_{v_i v_j}$ in the Fig.3 equal 0 because $F_{v_i} > T_{v_j}$ and the value

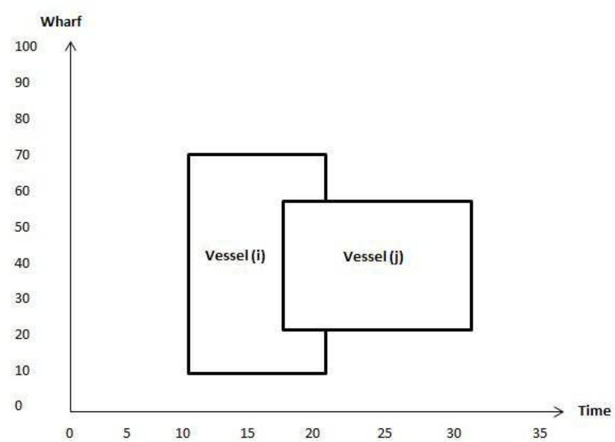

Figure 3: Illustration of time overlap between two vessels

of $\delta_{v_j v_i}$ in the same figure also equal 0 because $F_{v_j} > T_{v_i}$. That is mean there is an overlap in the time between these two vessels. Constraint (4) define $\sigma_{v_i v_j}$ such that $\sigma_{v_i v_j} = 0$ or 1 if the berthing position of vessel i plus the length of vessel i less than or equal the berthing position of vessel j; 0 if the berthing position of vessel i plus the length of vessel i greater than the berthing position of vessel j. The figures below illustrate how the value of $\sigma_{v_i v_j}$ is computed:

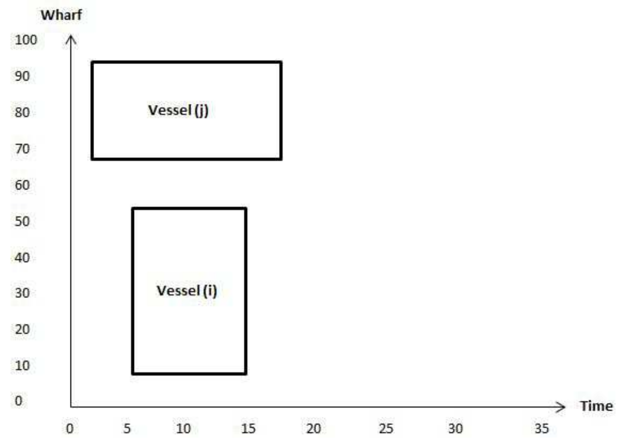

Figure 4: Illustration of no location overlap between two vessels

The value of $\sigma_{v_i v_j}$ in the Fig.4 equal 0 or 1 because $P_{v_i} + L_{v_i} \leq P_{v_j}$. Whereas, the value of $\sigma_{v_j v_i}$ in the same figure equal 0 because $P_{v_j} + L_{v_j} \geq P_{v_i}$. The value of $\sigma_{v_i v_j}$ in the Fig.5 equal 0 because

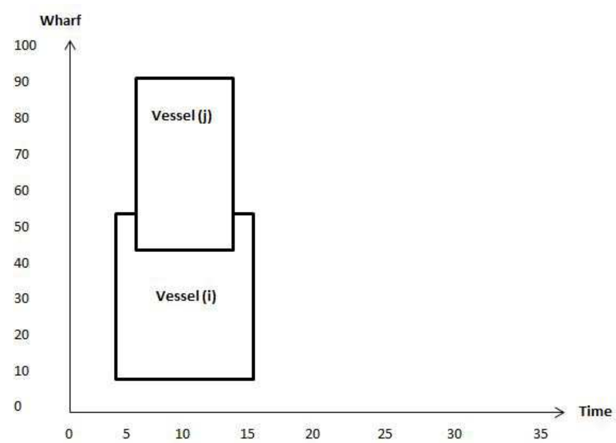

Figure 5: Illustration of location overlap between two vessels

$P_{v_i} + L_{v_i} > P_{v_j}$ and the value of $\sigma_{v_j v_i}$ in the same figure equal 0 because $P_{v_j} + L_{v_j} \geq P_{v_i}$. That is mean there is an overlap in the location between these two vessels. Constraint (5) ensures that the overlaps among vessels do not exist in the two dimensional (time and location) depending on the value of $\delta_{v_i v_j}$ and $\sigma_{v_i v_j}$.

Constraint (6) guarantees that vessels cannot moor before their arrivals. Constraint (7) implies that the berthing position plus the length of the vessel cannot exceed the range of the wharf.

The constraints (8-11) represent the main conditions for quay crane assignment. Constraint (8) and (9) respectively select the first and the last ships for each quay crane. Constraint (10) guarantees that ships are processed in a well-defined sequence. Constraint (11) guarantees each vessel is handled at least one quay crane.

The constraints (12-14) determine the starting time of quay cranes. Constraint (12) defines the starting time of the earliest vessel that is to be done by the q^{th} quay crane which should be after the ready time of the q^{th} quay crane. Note that the vessel v_o is a fake vessel. Constraint (13) states that the starting time of q^{th} quay crane on vessel v is no earlier than its berthing time if the q^{th} quay crane is assigned to the vessel. Constraint (14) ensures that the starting time of q^{th} quay crane on vessel v_j should be no earlier than finishing time of its vessel predecessor v_i.

Constraint (15) ensures that if a quay crane is allocated to a vessel, then it will start its processing from one task on that vessel. Constraint (16) ensures that if a quay crane is allocated to a vessel, then it will finish its processing from one task on that vessel. Constraint (17) shows a flow balance ensuring that tasks are performed in a well-defined sequence on every single vessel. Constraint (18) ensures that every task on each vessel must be completed by exactly one quay crane. Constraint (19) ensures that if a quay crane is not assigned to a vessel, the tasks on this vessel will not be performed by this quay crane. Constraint (20) simultaneously determines the completion time for each task. Constraint (21) defines the quay crane operation starting time. The completion time of each quay crane is computed by constraint (22). Constraint (23) determines the finishing time of each vessel.

When required, constraint (24) forces task i to be completed before task j for all the tasks which are in the set Φ. Constraint (25) defines Z_{ij}^v such that $Z_{ij}^v = 1$ when the operation of task j on vessel v starts after the operation for task i is completed; and 0 otherwise. Constraint (26) ensures that the pair of tasks that are members of the set Ψ will not be handled simultaneously.

Constraint (27), enforces interference avoidance among quay cranes. Suppose that tasks i and j are performed simultaneously and $l_i < l_j$. This means that $Z_{ij}^v + Z_{ji}^v = 0$. Note that both quay cranes and tasks are sorted in an increasing order of their relative location in the direction of increasing ship-bay numbers. Suppose furthermore that, for $q_1 < q_2$, quay crane q_1 performs tasks j and quay crane q_2 performs task i. Then, interference between quay cranes q_1 and q_2 results. However, in such a case, $\sum_{\theta=1}^{q_1}\sum_{\kappa=0}^{Nv} X_{\kappa j}^{\theta v} - \sum_{\theta=1}^{q_1}\sum_{\kappa=0}^{Nv} X_{\kappa i}^{\theta v} = 1$, it cannot be allowed because of constraint (27), and then we have $Z_{ij}^v + Z_{ji}^v = 0$.

Constraint (28) defines $\alpha_{b_i b_j}^{v_i v_j}$ such that $\alpha_{b_i b_j}^{v_i v_j} = 0$ or 1 if the berthing position of vessel v_i plus the location of task b_i on that vessel less than or equal the berthing position of vessel v_j plus the location of task b_j on that vessel; 0 if the berthing position of vessel v_i plus the location of task b_i on that vessel greater than the berthing position of vessel v_j plus the location of task b_j on that vessel. The figures below illustrate how the value of $\delta_{v_i v_j}$ is computed:

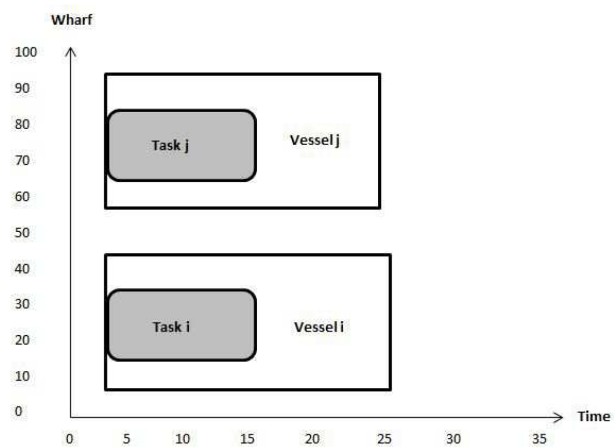

Figure 6: Illustration of no location overlap between two vessels

The value of $\alpha_{b_i b_j}^{v_i v_j}$ in the Fig.6 equal 0 or 1 because $P_{v_i} + l_{b_i}^v \leq P_{v_j} + l_{b_j}^v$. The value of $\alpha_{b_i b_j}^{v_i v_j}$ in the Fig.7 equal 0 because $P_{v_i} + l_{b_i}^v > P_{v_j} + l_{b_j}^v$. That is mean there is overlap in the position between these two tasks on these two vessels.

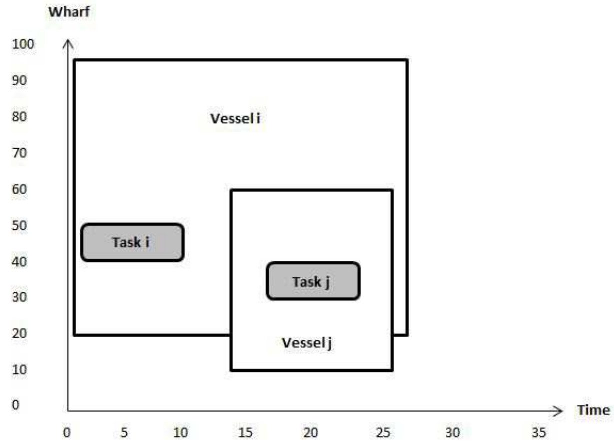

Figure 7: Illustration of location overlap between two vessels

Constraint (29) define $\beta^{v_i v_j}_{b_i b_j}$ such that $\beta^{v_i v_j}_{b_i b_j} = 0$ or 1 if the finishing time of task b_i on vessel v_i plus the processing time of task b_j on vessel v_j less than or equal the finishing time of task b_j on vessel v_j; 0 if the finishing time of task b_i on vessel v_i plus the processing time of task b_j on vessel v_j greater than the finishing time of task b_j on vessel v_j. The figures below illustrate how the value of $\beta^{v_i v_j}_{b_i b_j}$ is computed: The

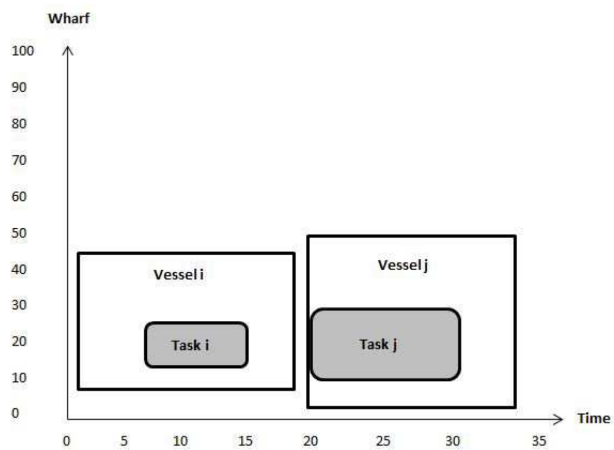

Figure 8: Illustration of no time overlap between two vessels

value of $\beta^{v_i v_j}_{b_i b_j}$ in the Fig.8 equal 0 or 1 because $D^{v_i}_{b_i} + p^{v_j}_{b_j} \leq D^{v_j}_{b_j}$. Whereas, the value of $\beta^{v_j v_i}_{b_j b_i}$ in the same figure equal 0 because $D^{v_j}_{b_j} + p^{v_i}_{b_i} > D^{v_i}_{b_i}$. The value of $\beta^{v_i v_j}_{b_i b_j}$ in the Fig.9 equal 0 because $D^{v_i}_{b_i} + p^{v_j}_{b_j} > D^{v_j}_{b_j}$ and the value of $\beta^{v_j v_i}_{b_j b_i}$ in the same figure equal 0 because $D^{v_j}_{b_j} + p^{v_i}_{b_i} > D^{v_i}_{b_i}$. That is mean there is overlap in the time between these two tasks on these two vessels. Constraint (30) prevents the interference among the quay cranes and the vessels depending on the value of $\beta^{v_i v_j}_{b_i b_j}$ and $\alpha^{v_i v_j}_{b_i b_j}$.

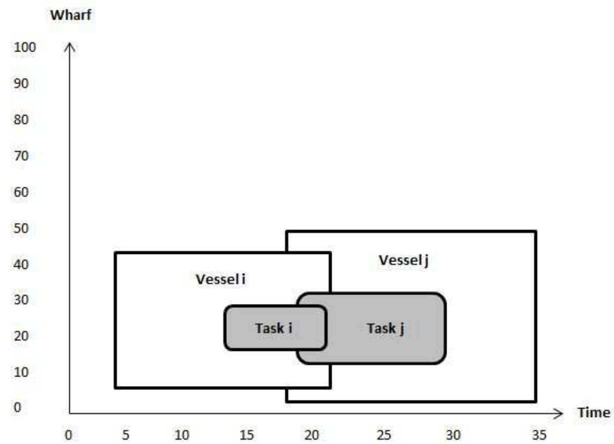

Figure 9: Illustration of time overlap between two vessels

5 Computational experiments

Numerical results have been obtained using CPLEX to solve instances of the suggested model which integrates Berth Allocation, Quay Crane Assignment, and Quay Crane Scheduling. 25 instances were randomly generated. They are described in Table 1. All experiments were carried out on a PC with Intel Core 2, 2.40 GHz CPU and 4 GByte RAM running Windows 7 operating system. CPLEX is limited to relatively small instances since large ones require astronomical CPU time. Unfortunately, such instances arise when considering large numbers of vessels, quay cranes, and task. Alternative approaches to solve practical instances of the model are being considered, [25, 27, 28, 24, 26].

Problem	No.Vessels	No.QCs	No.Tasks	No.Constraints	No.Variables	No.Int.Variables	CPLEX(h:m:s)
1	2	2	2	128	94	70	less than 1s
2	2	2	3	180	118	91	less than 1s
3	2	2	4	240	142	112	less than 1s
4	2	3	2	204	148	122	00.00.04
5	2	3	3	294	184	155	00.00.07
6	2	3	4	402	220	188	00.00.09
7	2	4	2	308	222	194	00.01.25
8	2	4	3	450	274	243	00.06.12
9	2	4	4	624	326	292	00.06.52
10	2	5	2	440	316	286	00.37.20
11	2	5	3	648	388	355	01.39.56
12	2	5	4	906	460	424	17.08.30
13	3	2	2	241	177	140	00.00.05
14	3	2	3	345	216	174	00.00.07
15	3	2	4	473	255	208	00.00.21
16	3	3	2	400	288	248	00.04.17
17	3	3	3	591	345	300	00.27.54
18	3	3	4	836	402	352	01.06.11
19	3	4	2	619	441	398	01.41.31
20	4	2	2	388	284	234	00.00.37
21	4	2	3	562	340	283	00.01.43
22	4	2	4	784	396	332	00.04.29
23	4	3	2	660	472	418	05.12.23
24	5	2	2	569	415	352	00.53.04
25	6	2	2	784	570	494	01.41.36

Table 1: Test problems solved by CPLEX in the experimental investigation

6 Conclusion

This paper describes an integrated mathematical model to simultaneously solve the seaside problems BAP, QCAP, and QCSP, that arise at container terminals. This approach is desirable because solving them individually and even when combined pairwise, may lead to suboptimal solutions. The problem is formulated as a mixed-integer programming model and then solved using B&B as implemented in CPLEX 12.6. Although the main contribution of this paper is this single extended model, the model itself has features which are not commonly represented in existing models. Our model is dynamic in that it copes with different ship arrival times. Quay crane interference avoidance is explicitly represented as constraints 27 when the quay cranes operate on the same vessel and as constraints 30 when quay cranes are on different vessels. Another advantage of this model is that it allows a quay crane that has become idle (finished its work) to move from its current vessel to another even if overall the current vessel to which it has been allocated is still being processed. Travel times of a quay crane between two holds on the same ship and between two holds on different vessels have also been considered.

We have experimented on relatively limited size cases with only few vessels, quay cranes and tasks. However, as can be seen, instances of the model grow to large sizes with hundreds of constraints and integer variables. This means that exact solution is computationally expensive. For instance, Problem 12 in Table 1, which involves two vessels, five quay cranes and four tasks, requires over 17 hours of CPU time using CPLEX. This problem may arise in a small provincial terminal port. Truly practical instances are beyond CPLEX, running on a PC. It is possible to reduce that time using simple parallelisation. However, brute force won't be enough on real-world instances of the model. For this reason, we are looking at enhancing the solution capability of the CPLEX B&B solver by using for instance combinatorial Benders' cuts, as well as alternative solution approaches such as evolutionary approaches and other Nature-inspired algorithms, [25, 27, 28, 24, 26].

References

[1] Christian Bierwirth and Frank Meisel. A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling*, 12(4):345–360, 2009.

[2] Christian Bierwirth and Frank Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615–627, 2010.

[3] Chun Yew Cheong, Mohamed Salahuddin Habibullah, Rick Siow Mong Goh, and Xiuju Fu. Multi-objective optimization of large scale berth allocation and quay crane assignment problems. In *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pages 669–676. IEEE, 2010.

[4] CY Cheong, KC Tan, DK Liu, and CJ Lin. Multi-objective and prioritized berth allocation in container ports. *Annals of Operations Research*, 180(1):63–103, 2010.

[5] SH Chung and King Lun Choy. A modified genetic algorithm for quay crane scheduling operations. *Expert Systems with Applications*, 39(4):4213–4221, 2012.

[6] Carlos F Daganzo. The crane scheduling problem. *Transportation Research Part B: Methodological*, 23(3):159–175, 1989.

[7] Mihalis M Golias, Maria Boile, and Sotirios Theofanis. Berth scheduling by customer service differentiation: A multi-objective approach. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):878–892, 2009.

[8] Yongpei Guan and Raymond K Cheung. The berth allocation problem: models and solution methods. *OR Spectrum*, 26(1):75–92, 2004.

[9] Akio Imai, Etsuko Nishimura, and Stratos Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research Part B: Methodological*, 35(4):401–417, 2001.

[10] Narges Kaveshgar, Nathan Huynh, and Saeed Khaleghi Rahimian. An efficient genetic algorithm for solving the quay crane scheduling problem. *Expert Systems with Applications*, 39(18):13108–13117, 2012.

[11] Kap Hwan Kim and Young-Man Park. A crane scheduling method for port container terminals. *European Journal of operational research*, 156(3):752–768, 2004.

[12] Der-Horng Lee, Hui Qiu Wang, and Lixin Miao. Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):124–135, 2008.

[13] Yusin Lee and Chuen-Yih Chen. An optimization heuristic for the berth scheduling problem. *European Journal of Operational Research*, 196(2):500–508, 2009.

[14] Pasquale Legato, Daniel Gullì, and Roberto Trunfio. The quay crane deployment problem at a maritime container terminal. In *Submitted to the 22th European Conference on Modelling and Simulation*, 2008.

[15] Chengji Liang, Jianquan Guo, and Yang Yang. Multi-objective hybrid genetic algorithm for quay crane dynamic assignment in berth allocation planning. *Journal of Intelligent Manufacturing*, 22(3):471–479, 2011.

[16] Jiyin Liu, Yat-wah Wan, and Lei Wang. Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics (NRL)*, 53(1):60–74, 2006.

[17] Wali Khan Mashwani and Abdellah Salhi. A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation. *Applied Soft Computing*, 12(9):2765–2780, 2012.

[18] Frank Meisel and Christian Bierwirth. Heuristics for the integration of crane productivity in the berth allocation problem. *Transportation Research Part E: Logistics and Transportation Review*, 45(1):196–209, 2009.

[19] Frank Meisel and Christian Bierwirth. A framework for integrated berth allocation and crane operations planning in seaport container terminals. *Transportation Science*, 47(2):131–147, 2013.

[20] Luigi Moccia, Jean-François Cordeau, Manlio Gaudioso, and Gilbert Laporte. A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics (NRL)*, 53(1):45–59, 2006.

[21] Su Nguyen, Mengjie Zhang, Mark Johnston, and Kay Chen Tan. Hybrid evolutionary computation methods for quay crane scheduling problems. *Computers & Operations Research*, 40(8):2083–2093, 2013.

[22] Etsuko Nishimura, Akio Imai, and Stratos Papadimitriou. Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*, 131(2):282–292, 2001.

[23] Roy I Peterkofsky and Carlos F Daganzo. A branch and bound solution method for the crane scheduling problem. *Transportation Research Part B: Methodological*, 24(3):159–172, 1990.

[24] JA Vázquez Rodríguez and A Salhi. A synergy exploiting evolutionary approach to complex scheduling problems. *Computer Aided Methods in Optimal Design and Operations, Series on Computers and Operations Research, World Scientific Publishing Co. Pvt. Ltd*, pages 59–68, 2006.

[25] José Antonio Vázquez Rodrıguez, Sanja Petrovic, and Abdellah Salhi. A combined meta-heuristic with hyper-heuristic approach to the scheduling of the hybrid flow shop with sequence dependent setup times and uniform machines. In *Proceedings of the 3rd Multidisciplinary International Conference on Scheduling: Theory and Applications. MISTA: Paris, France*, pages 506–513, 2007.

[26] José Antonio Vázquez RodrVguez and Abdellah Salhi. Hybrid evolutionary methods for the solution of complex scheduling problems. In *Advances in Artificial Intelligence*, pages 17–28. 2006.

[27] Abdellah Salhi and Eric S Fraga. Nature-inspired optimisation approaches and the new plant propagation algorithm. In *Proceedings of the ICeMATH2011*, pages K2–1–K2–8, 2011.

[28] Abdellah Salhi and José Antonio Vázquez Rodríguez. Tailoring hyper-heuristics to specific instances of a scheduling problem using affinity and competence functions. *Memetic Computing*, 6(2):77–84, 2014.

[29] Marcello Sammarra, Jean-François Cordeau, Gilbert Laporte, and M Flavia Monaco. A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling*, 10(4-5):327–336, 2007.

[30] Reza Tavakkoli-Moghaddam, A Makui, S Salahi, M Bazzazi, and F Taheri. An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering*, 56(1):241–248, 2009.

[31] Ching-Jung Ting, Kun-Chih Wu, and Hao Chou. Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications*, 41(4):1543–1550, 2014.

[32] Shan Tong. Genetic algorithm for dynamic berth allocation problem with discrete layout. In *Proceedings of the 2012 International Conference on Computer Application and System Modeling*. Atlantis Press, 2012.

[33] Chunxia Yang, Xiaojun Wang, and Zhenfeng Li. An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. *Computers & Industrial Engineering*, 63(1):243–253, 2012.

[34] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.