Quay crane scheduling problem with considering tidal impact and fuel consumption

Shucheng Yu[1], Shuaian Wang[2], Lu Zhen[1*]

[1] *School of Management, Shanghai University, Shanghai, China*

[2] *Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Abstract: This study investigates a quay crane scheduling problem with considering the impact of tides in a port and fuel consumption of ships. A mixed-integer nonlinear programming model is proposed. Some nonlinear parts in the model are linearized by approximation approaches. For solving the proposed model in large-scale problem instances, both a local branching based solution method and a particle swarm optimization based solution method are developed. Numerical experiments with some real-world like cases are conducted to validate the effectiveness of the proposed model and the efficiency of the proposed solution methods.

Keywords: port operations; container terminals; quay cranes; tide impacts; fuel consumption.

1. Introduction

The efficiency of port operations is vital for the competiveness of container terminals in the global maritime transportation network. The quay side is usually the bottleneck for increasing the efficiency of port operations. Quay cranes (QCs) are the most important and expensive resources at quay side of terminals. Thus it is an urgent and meaningful task to improve the QC scheduling efficiency for the practitioners in ports. There exists abundant literature on the QC scheduling problems (QCSPs). Most of them mainly investigated this problem through the perspective of port operators. However, there are two major players in container shipping: container shipping lines and container port operators. The benefits and cost issues on their sides (port operators and shipping lines) are intertwined. Efficient operations on one of the two sides will benefit the other side. For example, a scheduling decision of QCs will influence the departure time of a ship, which further influences the fuel consumption for the ship sailing to its next port. So we involve the factor of fuel consumption into the traditional QCSP so as to obtain a plan which benefits both port operators and shipping lines.

* Corresponding author. Email: lzhen@shu.edu.cn. Tel: +86-21-66137925. Fax: +86-21-66134284.

Moreover, this study considers the factors of the tide in the navigation channel on the basis of QC scheduling decisions. In reality, some mega-ports (e.g., Shanghai port, Antwerp port and Hamburg port) are tidal ports. Although berths and port basins of these ports are deep enough to moor mega-ships, the navigation channels are relatively shallow. The tide in a port fluctuates by following some patterns. Mega-ships need to take advantage of tide so as to pass through the channel. Thus the departure time of some mega-ships depends on the tide pattern, which is predicated in advance. If the tide height required for a ship to pass through the channel is not satisfied, the ship has to wait in the anchorage area for hours until the tide is high enough. The delay requires the ship to travel faster to reach its next port on schedule and higher speed leads to higher fuel consumption. Wang and Meng (2012) used the historical operating data from a global liner shipping company to elaborate on the relationship between sailing speed and fuel consumption. The diagram of the tide pattern in a port is usually posted on the wall of the planning office of the port, and acts as one of the most important reference information for making the daily berth planning and QC scheduling decisions in the port.

By combining the above two factors, this study conducts an explorative study on a QC scheduling problem with considering the impact of tides in a port and fuel consumptions for arrival ships. We propose a mixed-integer nonlinear programming model. Some nonlinear parts in the model are then linearized by some approximation approaches. In addition, both a local branching based solution method and a particle swarm optimization based solution method are suggested to solve the proposed model in large-scale problem instances. Numerical experiments with some real-world like cases are conducted to validate the effectiveness of the proposed model and the efficiency of the proposed solution methods.

The remainder of this paper is organized as follows. Section 2 is the relevant literature review. Section 3 introduces the problem background of this paper. A mixed-integer nonlinear programming model is developed and some approximation approaches is introduced in Section 4. Section 5 analyzes the structure of the problem and introduces the solution methods. Section 6 shows an experimental analysis of the model. Conclusions are presented in Section 7.

2. Literature review

QCs are critical machines loading and unloading containers onto and from vessels. The QCSP is one of the most studied operational problems in container terminals. The QCSPs have received abundant attention since the publication of the papers of Daganzo (1989), Peterkofsky and Daganzo (1990). In their works, static and dynamic QCSP for multiple vessels were considered. A MIP was developed to minimize the total weighted departure time of vessels. A Branch-and-Bound (B&B) method was proposed by Peterkofsky and Daganzo (1990). However, both studies failed to consider the spatial interference.

The QCSPs can be classified into two categories, i.e., the bay based QCSPs and the cluster based QCSPs. The two studies of Daganzo (1989), Peterkofsky and Daganzo (1990) belong to the bay based QCSPs. Zhu and Lim (2006) studied the bay-based QCSP, and considered the non-crossing spatial constraints. They assumed the ships can be divided into 'holds' so that QCs can work from one hold to another. A B&B algorithm was used to obtain the optimal solution. Lim et al. (2007) modeled the bay-based QCSP as an *m*-parallel machine scheduling problem. Kim and Park (2004) proposed the cluster based QCSP. They defined a 'task' as discharging or loading operations for a group of containers in their model formulation. The goal of the QCSP becomes to determine the sequence of the discharging and loading operations and the precedence relationships among the clusters are considered. Their study provided a comparison between the B&B algorithm and the greedy randomized adaptive search procedure (GRASP). Moccia et al. (2006) strengthened the constraints proposed by Kim and Park (2004). They added the idle times originated from the interactions between QCs. A revised QCSP model was formulated. Bierwirth and Meisel (2009) designed a heuristic solution procedure based on the B&B to solve the model proposed by Kim and Park (2004). Then based on the work of Bierwirth and Meisel (2009), Legato et al. (2012) provided a so called rich model for QC scheduling that covered important issues of practical relevance. They extended the B&B scheme by revising and extending the lower bounds and branching criteria. Recently, studies on the QCSPs have focused on considering some realistic constraints and solving large scale problems in practice (Bierwirth and Meisel, 2010). The realistic constraints mainly contain: (1) the safety distance and non-crossing constraints (Liu et al., 2006; Lee et al., 2008; Bierwirth and Meisel, 2009); (2) QC travel time between bays (Kim and Park, 2004; Ng and Mak, 2006; Tavakkoli et al., 2009); (3) precedence relationship between tasks and QC ready times (Bierwirth and Meisel, 2009; Meisel and

Bierwirth, 2011; Chen et al., 2014). In addition, some uncertain issues are considered in some port operation related studies (Zhen, 2015).

Although there are a large number of studies on the QCSPs since the 1990s, few papers on the QCSPs have considered fuel consumption and tidal impacts. The factors of fuel consumption and emissions have been considered in some other maritime transportation related studies. Golias et al. (2009) regarded the arrival time of vessels as decision variables. They presented a berth-scheduling policy to minimize vessel delayed departures and indirectly reduce fuel consumption and emissions produced by the vessels in their idle mode. But they ignored an important fact that the sailing emissions are more important than the emissions in the idle mode. Du et al. (2011) proposed a novel model on berth allocation considering fuel consumption. They considered both the vessel emission in sailing periods and vessel emissions in mooring periods through a post-optimization phase on waiting time. A berth allocation strategy as well as a second-order cone optimization based solution algorithm was proposed. Hu et al. (2014) discussed the cooperative strategies between port operators and shipping lines. They considered the effects of QCs on operating cost and vessels' fuel consumption and emissions by optimizing sailing speed. A novel nonlinear multi-objective MIP model was developed to consider vessels' fuel consumption and emissions. Some related papers have addressed the economic aspects (e.g., profit, transit time, fuel consumption cost) of vessel routing, deployment in the linear shipping (Ronen, 2011; Wang and Meng, 2012; Meng et al., 2013) while considering the fuel consumption in their optimization models. The fuel consumption is very sensitive to the sailing speed. Many studies focused on the impact analysis of sailing speed on bunker consumption. Ronen (1982) analyzed the tradeoff between fuel savings through slow steaming and the loss of revenues due to the resulting voyage extension.

For the tidal factor, most seaside operation scheduling problems did not consider the tide impacts (Fagerholt, 2004) or the time variation in draft constraints (Rakke et al., 2011; Song and Furman, 2013). In reality, the time-varying draft restrictions should be considered especially for the ports whose water depths of the navigation channels are inadequate for all of the vessels. Du et al. (2015) quantified the impacts of tides on seaside operations in container ports and reformulated the berth allocation problem by modeling tidal impacts. They took a planning horizon of several days involving several high/low tides and integrated the tide window constraints into the berth allocation problem. Some other studies about the tide impacts on container seaports can be found in Qureshi et al. (2009) and Xu et al. (2012).

Based on the above existing literature, this study extends the traditional QCSP by considering the ship fuel consumption and the tidal factor. A nonlinear MIP model is developed. The proposed model simplifies a well-known QCSP model (Bierwirth and Meisel, 2009) by removing the two dummy tasks. In addition, the proposed model not only synchronizes the QCSP and fuel consumption problem by borrowing the idea of Wang and Meng (2012), but also integrates specific tide window constraints into the QCSP by considering the work of Du et al. (2015). For solving the proposed model, this study develops two types of new solution methods, which may also contribute to the existing solution methods for the QCSPs.

3. Problem description

Consider liner container shipping companies that operate a number of ships. When a ship completed its QC handling process, the ship must reach next port on time or before its planned time. If the departure time is late, it requires the ship to travel faster and higher speed leads to higher fuel consumption. Some mega-ships with deep-draft can only leave a port when the sea level is high enough. Therefore tide impacts the departure time of ships, especially tide-dependent ships.

3.1 Quay crane scheduling problem

QCSP belongs to one type of task (job) scheduling problem. In the context of port operations, a task (job) could be an activity for loading or unloading a group of containers between the quay side and yard side. And the resources for handling these tasks (jobs) are QCs that are the most expensive equipment at terminals. For a given number of ships, tasks and QCs. The QCSP is to assign the QCs to ships and determine the sequence of tasks for each QC in order to minimize the make-span of the whole seaside operation time. As the QCSP is a well-studied area, this section would not elaborate on the background of the traditional QCSP. A toy example is provided to illustrate the problem and its solutions. Table 1 shows data for a small QCSP instance which contains seven tasks placed in two ships with seven bays, in which P_i denotes QC processing time of task i. l_{0q} is the initial position of QC q. r_{0q} represents the earliest available time of QC q. g denotes the QC's unit travel time per bay and $\Phi(i,j)$ denotes the precedence constraint that task i must be handled before task j.

A simple QC assignment plan is shown in Figure 1(a). In the plan, one QC is assigned to ship 1 and another QC is assigned to the second ship. One QC's sequence of tasks is $2 \rightarrow 1 \rightarrow 3 \rightarrow 4$ and the other QC's sequence is $5 \rightarrow 6 \rightarrow 7$, respectively. While Figure 1(b) shows the optimal solution, in which the QC2 is assigned to serve two ships. The whole operation time reduces from 24 to 21.

Table 1: A toy QCSP example

Task index i	1	2	3	4	5	6	7
Processing time P_i	4	8	5	4	3	8	3
Bay position l_i	1	1	3	4	5	5	6
Assign to ship	ship1	ship1	ship1	ship1	ship2	ship2	ship2
Precedence constraint tasks	$\Phi = \{(2,1)\}$						
QC1	$l_{01} = 1$, r_{01}=0						
QC2	$l_{02} = 4$, r_{02}=0						
QC unit travel time	$g = 1$						

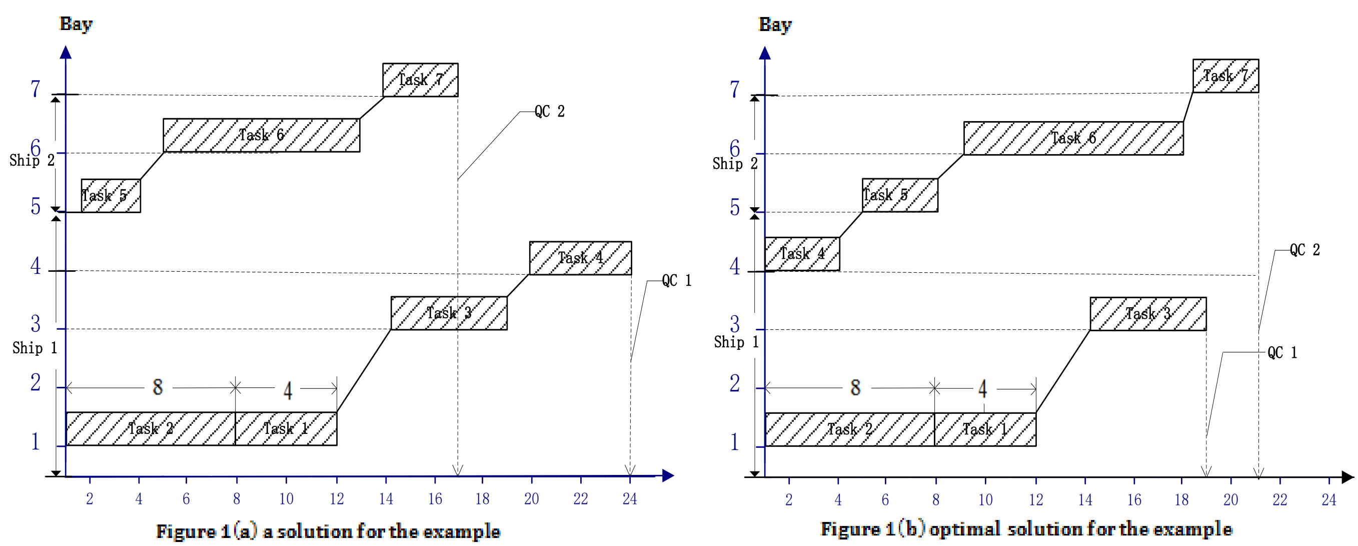


Figure 1: An example about the quay crane scheduling problem

Given a set of tasks Ω= $\{1, \cdots, N\}$, and a set of QCs $Q = \{1,2,\cdots,|Q|\}$ serving a set of ships $\mathcal{H}$=$\{1, 2, \cdots,|\mathcal{H}|\}$. Each task i denotes a loading or unloading process of a container group. Each task has a processing time p_i and a bay position l_i. In order to define the whole completion time to each ship. We establish a corresponding relationship between the completion time of each ship and the completion time of its tasks. Let Ω_τ denote the set of tasks for Ship τ, i.e., Ω_τ=$\{1,\cdots, N_\tau\}$. Let t_τ denote the QC processing completion time of Ship τ. Then t_τ can be calculated by the final completion task of Ship τ. Moreover, the precedence relationship may exist between the pairs of tasks located in the same bay. We define Φ as the precedence constraint for task pairs. Let Ψ represents the set of tasks that cannot be processed simultaneously hence $\Psi \supseteq \Phi$. Some practical constraints such as at most one QC handled at a bay anytime, QCs cannot be allowed to cross each other and the adjacent QCs should have to keep a safety margin δ that is measured in units of bays are all considered in QCSP.

3.2 Impact of tides on ships

Many container ports have been enduring influenced by tides, especially for some mega-ports, such as Shanghai Guandong International Container Terminal (SGICT) in the Waigaoqiao area, the port of Tampa on the northern shore of Hillsborough Bay, the international seaport of Antwerp in Belgium and the port of Hamburg in Europe. In the Antwerp port about seven out of ten vessels were influenced by tide influence before dredging of the Scheldt River in 2010. The tide in the port let the depth of the navigation channel fluctuate is about 4.2m~4.5m (Port of Antwerp 2016). About 100 years ago, the Waigaoqiao area of Shanghai port has established the earliest accurate tidal predication system in China. Since dredged material is deposited the Waigaoqiao area, the 10.5m-13.0m deep Yangtze River dredged shipping channel cannot serve the deep-drafts ships navigate this channel unless the tide is sufficiently high (SHMAS, 2016).

Those ports face big challenges in berth planning due to the tidal effects. Though the berth position is deep enough to accommodate ships, the periodically changing water depth increases the difficulty of serving ships especially for some deep-drafted ships. So a suitable berth and departure time should be carefully determined with considering the influence of tide. The influence of tide impact of ship's berthing and departure time is shown in Figure 2.

As shown in Figure 2, Ship τ completes its QC handling operation at a point time t_τ, which is in the infeasible time window $[T_\tau^1, T_\tau^2]$. Hence the Ship τ cannot pass through the channel due to the low sea level of the tide. Considering the time through the navigation channel, Ship τ has to wait at the anchorage ground if its QC handling operation time t_τ lies in the time window $[T_\tau^1 - N^{out}, T_\tau^2]$, where N^{out} is the time that Ship τ goes through the navigation channel. For the interest of simplicity, we assume N^{out} is identical for all the ships. Then we can define the infeasible time windows as $[T_\tau^{2n-1} - N^{out}, T_\tau^{2n}]$ in the planning horizon for the Ship τ. If the time when Ship τ completes its QC handling is in an infeasible time window $[T_\tau^{2n-1} - N^{out}, T_\tau^{2n}]$, the time d_τ when Ship τ departs the port can be calculated as $d_\tau = T_\tau^{2n} + N^{out}$.

On the other hand, if the processing completion time t_τ just locates in the feasible time window $[T_\tau^{2n}, T_\tau^{2n+1} - N^{out}]$, which means the tide has risen to an adequate height that allows Ship τ to sail out of the navigation channel freely, the departure time is $d_\tau = t_\tau + N^{out}$. In this study, we mainly consider the tide's influence on the Ships' departure time.

To model the impact of tides on the departure time window, first we divide the ships into two categories, and let $\mathcal{H}^{\text{Tide}}$ and $\mathcal{H} - \mathcal{H}^{\text{Tide}}$ denote the set of ships whose berthing/departing time are tide dependent and tide independent, respectively. LT_τ denotes the time windows when the sea level is below the minimum required level for the Ship τ, i.e., $LT_\tau = \{[T_\tau^1 - N^{out},\ T_\tau^2], \cdots, [T_\tau^{2n-1} - N^{out}, T_\tau^{2n}],\ n \in \{1,2,\cdots D\}\}$. Here $[T_\tau^{2n-1} - N^{out}, T_\tau^{2n}] = [T_\tau^1 + 12 \times (n-1) - N^{out}, 12 \times (n-1)]$, D is the number of the time windows (tide cycles) considered in the planning horizon, n is the index of the time windows. The tide is periodic and the period is twelve hours. For each Ship $\tau \in \mathcal{H}^{\text{Tide}}$, we have:

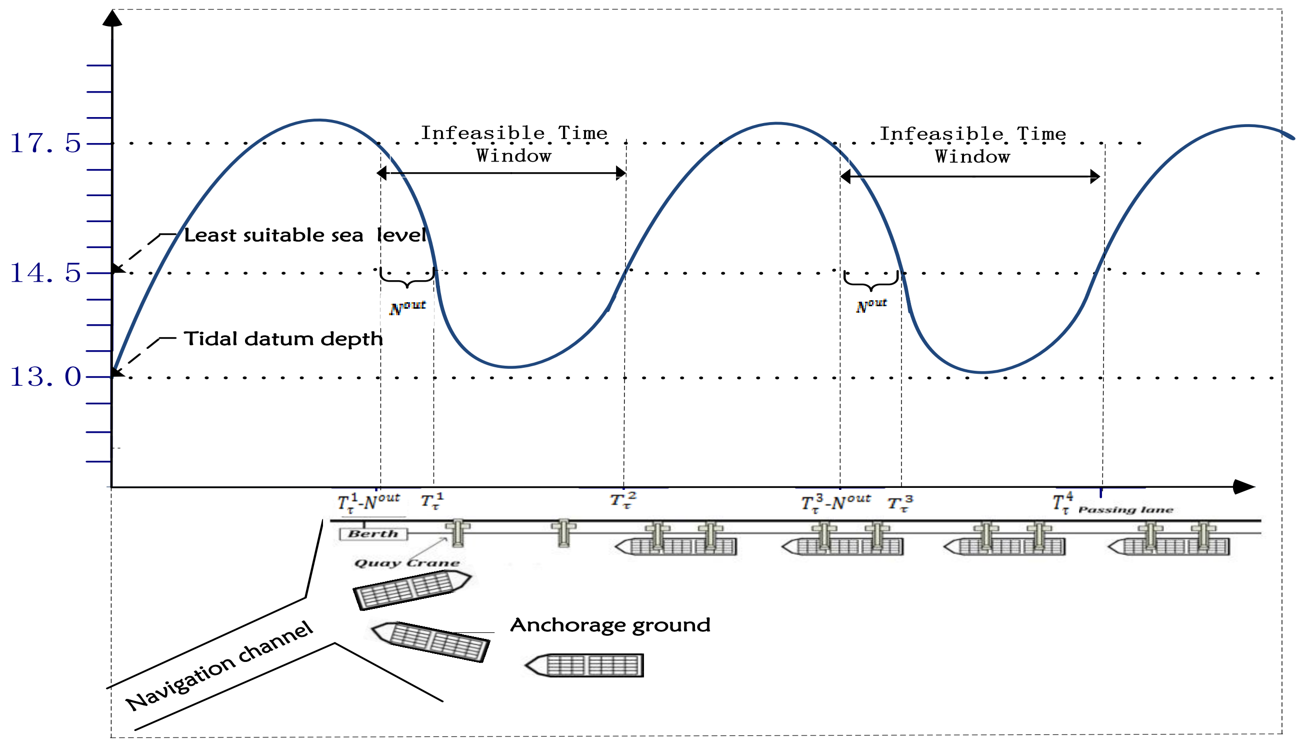


Figure 2: Berthing and departure time with tidal influence

(1) If $t_\tau \notin LT_\tau$ and $t_\tau + N^{out} \notin LT_\tau$, which denotes Ship τ completes its QC processing time located in the feasible time window and passes through the navigation channel during the

feasible time window. The departure time of Ship τ can be calculated as $d_\tau = t_\tau + N^{out} \in [T_\tau^{2n}, T_\tau^{2n+1} - N^{out}], \forall n \in \{1,2,\cdots D\}$.

(2) If $t_\tau \notin LT_\tau$ and $t_\tau + N^{out} \in LT_\tau$, which means Ship τ completes its QC processing time located in the feasible time window while Ship τ does not have enough time to pass through the navigation channel during the feasible time window. The ship has to wait at the anchorage area till the next feasible time window, so the departure time $d_\tau = T_\tau^{2n} + N^{out} \in [T_\tau^{2n}, T_\tau^{2n+1} - N^{out}], \forall n \in \{1,2,\cdots D\}$.

(3) If $t_\tau \in LT_\tau$, which indicates ship τ cannot pass through the channel due to the low sea level of the tide, then the departure time d_τ can be expressed as $d_\tau = T_\tau^{2n} + N^{out} \in [T_\tau^{2n}, T_\tau^{2n+1} - N^{out}], \forall n \in \{1,2,\cdots D\}$.

Based on the above discussions, we can calculate the departure time of Ship τ as follows.

$$d_\tau = \begin{cases} t_\tau + N^{out}, & \forall\ \tau \in \mathcal{H} - \mathcal{H}^{\text{Tide}} \\ t_\tau + N^{out}, & \forall\ \tau \in \mathcal{H}^{\text{Tide}}, t_\tau \notin LT_\tau, t_\tau + N^{out} \notin LT_\tau \\ T_\tau^{2n} + N^{out}, & \forall \tau \in \mathcal{H}^{\text{Tide}}, t_\tau \notin LT_\tau, t_\tau + N^{out} \in LT_\tau \\ T_\tau^{2n} + N^{out}, & \forall\ \tau \in \mathcal{H}^{\text{Tide}}, t_\tau \in LT_\tau \end{cases} \quad (1)$$

In the branch conditions of Constraints (1), t_τ is a decision variable in our model, which reduces the modeling capacity of the constraints. To eliminate decision variables from the conditions of Constraints (1), two auxiliary binary variables E_τ^n and F_τ^n are introduced for expressing each infeasible tide time window and feasible tide time window of Ship $\tau \in \mathcal{H}^{\text{Tide}}$, respectively.

$$E_\tau^n = \begin{cases} 1, & if \quad t_\tau \in (T_\tau^{2n-1} - N^{out}, T_\tau^{2n}] \\ 0 & otherwise \end{cases} \quad \forall n \in \{1,2,\cdots D\}$$

$$F_\tau^n = \begin{cases} 1, & if \quad t_\tau \in (T_\tau^{2n}, T_\tau^{2n+1} - N^{out}] \\ 0 & otherwise \end{cases} \quad \forall n \in \{1,2,\cdots D\}$$

in which $E_\tau^n = 1$ means t_τ lies in one of the infeasible tide time window $(T_\tau^{2n-1} - N^{out}, T_\tau^{2n}]$, where $n \in \{1,2,\cdots D\}$. $F_\tau^n = 1$ means t_τ lies in one of the feasible tide time windows $(T_\tau^{2n}, T_\tau^{2n+1} - N^{out}]$, where $n \in \{1,2,\cdots D\}$. Then, Constraints (1) can thus be further defined as follows:

$$d_\tau = t_\tau + N^{out} \qquad \forall\ \tau \in \mathcal{H} - \mathcal{H}^{\text{Tide}} \tag{2}$$

$$(T_\tau^{2n-1} - N^{out}) \times E_\tau^n \le t_\tau \le T_\tau^{2n} + M \times (1 - E_\tau^n) \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{3}$$

$$\textstyle\sum_{n=1}^{D} E_\tau^n \le 1 \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}} \tag{4}$$

$$T_\tau^{2n} \times F_\tau^n \le t_\tau \le T_\tau^{2n+1} - N^{out} + M \times (1 - F_\tau^n) \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{5}$$

$$\textstyle\sum_{n=1}^{D} F_\tau^n \le 1 \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}} \tag{6}$$

$$(t_\tau + N^{out}) \times F_\tau^n \le d_\tau \le t_\tau + N^{out} + M \times (1 - F_\tau^n) \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{7}$$

$$(T_\tau^{2n} + N^{out}) \times E_\tau^n \le d_\tau \le T_\tau^{2n} + N^{out} + M \times (1 - E_\tau^n) \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{8}$$

$$\textstyle\sum_{n=1}^{D} E_\tau^n + \sum_{n=1}^{D} F_\tau^n = 1 \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}} \tag{9}$$

$$E_\tau^n, F_\tau^n \in \{0,1\} \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{10}$$

The explanatory details of Constraints (2)−(10) can be found in section 4.1.

3.3 Fuel consumption

Generally speaking, the fuel consumption on one hand depends on the design and structure of a ship. On the other hand the fuel consumption is sensitive to its sailing speed. If the ship's structure and size is determined, the fuel consumption per unit time for the ship mainly depends on its sailing speed.

Based on the suggestions of MAN Diesel and Turbo (2004), Wang and Meng (2012), we let $g(v_\tau)$ represent the daily fuel consumption (tons) function of Ship τ for a sailing speed v_τ.

The power function can be calculated as

$$g(v_\tau) = a \times (v_\tau)^b,\ a > 0 \tag{11}$$

We define $Q(v_\tau)$ (tons / nautical mile) denote the fuel consumption per nautical mile at the speed of v_τ. Then the total fuel consumption of all the ships to sail to the next ports can be calculated by:

$$\sum_{\tau=1}^{|\mathcal{H}|} Q(v_\tau) \times L_\tau \tag{12}$$

where L_τ denotes the voyage distance (nautical mile) of Ship τ from the current port to its next port and $\mathcal{H}$ is the set of all the ships.

4. Model formulation

The following three subsections are structured as follows. A mathematical model for the QCSP is presented in subsection 4.1. Some alternatives decision variables and convexity of objective function are introduced in subsection 4.2. The approximation method used for the non-linear objective function is elaborated in the third subsection.

4.1 *Notations*

Index and sets:

q	index of QCs
i, j	index of tasks for QC handling
τ	index of ships that visit the current port
$\mathcal{H}^{\text{Tide}}$	set of ships that are tide dependent
$\mathcal{H}$	set of all the ships ; $\mathcal{H}=\{1,2,\cdots,|\mathcal{H}|\}$
Ω_τ	set of tasks for Ship τ, $\Omega_\tau=\{1, \cdots, |\Omega_\tau|\}$
Ω	set of tasks for all the ships, $\Omega=\Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_{|\mathcal{H}|}$
Q	set of QCs, $Q = \{1,2,\cdots,|Q|\}$, these QCs are numbered sequentially along the quay from one side to the other
LT_τ	time windows when the sea level is below the minimum required sea level for Ship τ, $LT_\tau = \{[T_\tau^1 - N^{out}, T_\tau^2], [T_\tau^3 - N^{out}, T_\tau^4], \cdots, [T_\tau^{2n-1} - N^{out}, T_\tau^{2n}]\}, n = 1,2,\cdots D$.

Here, D is the number of time windows (tide cycles) considered in the planning horizon

Φ set of task pairs (i,j) in the same bay that task i must precede task j when they are handled by QCs

Ψ set of all task pairs that cannot be performed simultaneously with considering the safety QC margin constraint and task precedence constraint, $\Phi \subseteq \Psi$

Parameters:

l_i index of the bay, which task i is related to

l_{0q} initial position of QC q

p_i QC processing time of task i

r_{0q} earliest available time of QC q

v_τ^{min} minimum speed of Ship τ

$v_\tau^{\max}$ maximum speed of Ship τ

L_τ voyage distance (nautical mile) for Ship τ sailing from the current port to its next port

N^{out} time for ships passing through the navigation channel

S_τ planned time for Ship τ to arrive at its next port

M a sufficiently large positive number

Variables:

c_i completion time of task i

d_τ departure time for Ship τ leaving the current port

t_τ time when QC handling activities are finished for Ship τ

v_τ sailing speed of Ship τ from the current port to its next port

x_{iq} set to 1 if task i is processed by QC q, and 0 otherwise

z_{ij} set to 1 if task j starts after the completion of task i, and 0 otherwise

E_τ^n set to 1 if $t_\tau \in (T_\tau^{2n-1} - N^{out}, T_\tau^{2n}], n = 1,2,\cdots D$, and 0 otherwise

F_τ^n set to 1 if $t_\tau \in (T_\tau^{2n}, T_\tau^{2n+1} - N^{out}], n = 1,2,\cdots D$, and 0 otherwise

Mathematical model:

$(\boldsymbol{\mathcal{M}\_1})$ *Minimize* $\sum_{\tau=1}^{|\mathcal{H}|} Q(v_\tau) \times L_\tau$

(13)

s.t. $\sum_{q \in Q} x_{iq} = 1 \qquad \forall\ i \in \Omega$ (14)

$$r_{0q} + g \times \left| l_{0q} - l_i \right| + p_i - c_i \leq M \times (1 - x_{iq}) \quad \forall \; i \in \Omega, \forall q \in Q \tag{15}$$

$$c_i \leq c_j - p_j \quad \forall \; (i,j) \in \Phi \tag{16}$$

$$z_{ij} + z_{ji} = 1 \quad \forall \; (i,j) \in \psi \tag{17}$$

$$c_i + p_j - c_j \leq M \times (1 - z_{ij}) \quad \forall \; i,j \in \Omega \tag{18}$$

$$c_j - p_j - c_i \leq M \times z_{ij} \quad \forall \; i,j \in \Omega \tag{19}$$

$$x_{iv} + x_{jw} \leq 1 + z_{ij} + z_{ji} \quad \forall \; (i,j,v,w) \in \Theta \tag{20}$$

$$c_i + \Delta_{ij}^{vw} + P_j - C_j \leq M \times \left(3 - z_{ij} - x_{iv} - x_{jw}\right) \quad \forall \; (i,j,v,w) \in \Theta \tag{21}$$

$$c_j + \Delta_{ij}^{vw} + P_i - C_i \leq M \times \left(3 - z_{ji} - x_{iv} - x_{jw}\right) \quad \forall \; (i,j,v,w) \in \Theta \tag{22}$$

$$t_\tau \geq c_i \quad \forall \; i \in \Omega_\tau, \tau \in \mathcal{H} \tag{23}$$

$$d_\tau = t_\tau + N^{out} \quad \forall \; \tau \in \mathcal{H} - \mathcal{H}^{\mathrm{Tide}} \tag{24}$$

$$(T_\tau^{2n-1} - N^{out}) \times E_\tau^n \leq t_\tau \leq T_\tau^{2n} + M \times (1 - E_\tau^n) \quad \forall \; \tau \in \mathcal{H}^{\mathrm{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{25}$$

$$\textstyle\sum_{n=1}^{D} E_\tau^n \leq 1 \quad \forall \; \tau \in \mathcal{H}^{\mathrm{Tide}} \tag{26}$$

$$T_\tau^{2n} \times F_\tau^n \leq t_\tau \leq T_\tau^{2n+1} - N^{out} + M \times (1 - F_\tau^n) \quad \forall \; \tau \in \mathcal{H}^{\mathrm{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{27}$$

$$\textstyle\sum_{n=1}^{D} F_\tau^n \leq 1 \quad \forall \; \tau \in \mathcal{H}^{\mathrm{Tide}} \tag{28}$$

$$(t_\tau + N^{out}) \times F_\tau^n \leq d_\tau \leq t_\tau + N^{out} + M \times (1 - F_\tau^n) \quad \forall \; \tau \in \mathcal{H}^{\mathrm{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{29}$$

$$(T_\tau^{2n} + N^{out}) \times E_\tau^n \leq d_\tau \leq T_\tau^{2n} + N^{out} + M \times (1 - E_\tau^n) \quad \forall \; \tau \in \mathcal{H}^{\mathrm{Tide}}, \forall n \in \{1,2,\cdots D\} \tag{30}$$

$$\sum_{n=1}^{D} E_\tau^n + \sum_{n=1}^{D} F_\tau^n = 1 \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}}$$

(31)

$$\frac{L_\tau}{v_\tau} \le S_\tau - d_\tau \qquad \forall\ \tau \in \mathcal{H}$$

(32)

$$v_\tau^{min} \le v_\tau \le v_\tau^{\max} \qquad \forall\ \tau \in \mathcal{H}$$

(33)

$$E_\tau^n, F_\tau^n \in \{0,1\} \qquad \forall\ \tau \in \mathcal{H}^{\text{Tide}}, \forall n \in \{1,2,\cdots D\}$$

(34)

$$x_{iq}, z_{ij} \in \{0,1\} \qquad \forall\ i,j \in \Omega, \forall q \in Q \qquad (35)$$

Objective (13) is to minimize the total fuel consumption of all the ships during the voyages from the current port to their next ports. Constraints (14) ensure that each task should be handled one and only one QC. Constraints (15) restrict the earliest starting time of each QC. Constraints (16) guarantee that if task i and task j belong to the set $\Phi(i,j)$ in the same bay, task i must precede task j in the QC handling process. Constraints (17) ensure if task i and task j belong to the set $\psi(i,j)$, the two tasks cannot be performed simultaneously. Constraints (18) and Constraints (19) define the binary variable z_{ij}, which equals one in the case that task j must be handled after the operation of task i and equals zero in the case that task i starts after the completion of task j. Constraints (20)−(22) consider the non-crossing constraints and safety margin constraints if QCs v and w are adjacent and their processing tasks are i and j, respectively (Bierwirth and Meisel, 2009). Constraints (23) connect the completion time of tasks (c_i) with the completion time (d_τ), which is the handling completion time for Ship τ. Constraints (24) define the departure time of d_τ if the ships are tide-independent. Constraints (25)−(26) bound the auxiliary decision variable E_τ^n, if the tide dependent Ship τ's QC processing completion time t_τ lies in the infeasible time window $[T_\tau^{2n-1} - N^{out}, T_\tau^{2n}], n = 1,2,\cdots D$. Similarly, Constraints (27)−(28) bound the auxiliary decision variable F_τ^n, if the tide dependent Ship τ's QC processing completion time t_τ lies in the feasible time window $[T_\tau^{2n}, T_\tau^{2n+1} - N^{out}], n = 1,2,\cdots D$. Constraints (29)−(30) define the departure time of d_τ if the Ship τ is tide-dependent. Constraints (31) enforce the Ship τ's QC processing completion time t_τ either lies in the infeasible time window or in the feasible time window, if Ship τ is tide dependent. Constraints (32) enforce each ship must reach next port on time or before planned time. Constraints (33) define the speed variability of each ship. Constraints (34)−(35) define the binary variables.

4.2 Alternative decision variables and convexity of objective function

We define a new decision variable to represent the reciprocal of sailing speed v_τ, denoted by u_τ, as the new decision variable, namely,

$$u_\tau = 1/v_\tau, \quad \forall \ \tau \in \mathcal{H}, \tag{36}$$

then, Constraints (32) turns to be

$$L_\tau \times u_\tau = S_\tau - d_\tau \quad \forall \ \tau \in \mathcal{H} \tag{37}$$

Constraints (33) can be written as

$$1/v_\tau^{\max} = u_\tau^{\min} \le u_\tau \le \frac{1}{v_\tau^{min}} = u_\tau^{\max} \quad \forall \ \tau \in \mathcal{H}, \tag{38}$$

where $u_\tau^{\min}$ and $u_\tau^{\max}$ denote the minimum and maximum sailing speed of u_τ, respectively. The fuel consumption function $g(v_\tau)$ can be expressed as a function of the reciprocal of sailing speed. We define

$$Q(u_\tau) = g(1/u_\tau) \quad \forall \ \tau \in \mathcal{H} \tag{39}$$

Hence, the model ($\boldsymbol{\mathcal{M}\_1}$) is equivalent to the following model by rewriting objective function shown in Eq. (12) as a function of the alterative decision variables.

$$(\boldsymbol{\mathcal{M}\_2}) \quad Minimize \sum_{\tau=1}^{|\mathcal{H}|} Q(u_\tau) \times L_\tau \tag{40}$$

s.t. Constraints (14)−(31), (34) −(35) , (37)− (38).

Note that ($\boldsymbol{\mathcal{M}\_2}$) is a MIP model with nonlinear objective function (40). We exploit the special structure of ($\boldsymbol{\mathcal{M}\_2}$) and prove Objective (40) is convex. As the daily fuel consumption (tons) of Ship τ is $a \times (v)^b$, $a > 0$. Note that $Q(u_\tau)$ represents the fuel consumption (tons) per nautical mile, we have

$$Q(u_\tau) = g\left(\frac{1}{u_\tau}\right) = \frac{a \times (1/u_\tau)^b}{24 \times (1/u_\tau)} = a \times (u_\tau)^{1-b}/24 \quad \forall \ \tau \in \mathcal{H} \tag{41}$$

As the function $Q(u_\tau)$ is convex in the interval $[u_\tau^{\min}, u_\tau^{\max}]$, hence Objective (40) is convex. Wang and Meng (2012) considered the coefficient b is between 2.7 and 3.3, a is between

0.004 and 0.037. In addition, Ronen (1982) mentioned a proper value of b is 3. We set a =0.02, b=3 in the numerical experiments of this study.

4.3 Approximation of the non-linear objective functions

Although Objective (40) is nonlinear, it is convex. For handling this nonlinear part, a strategy for constructing linear relaxations of mixed-integer nonlinear programs can be used (Tawarmalani and Sahinidis, 2004). We use an outer-approximation manner based on tangent lines. The outer-approximation method adds linear constraints to the objective function.

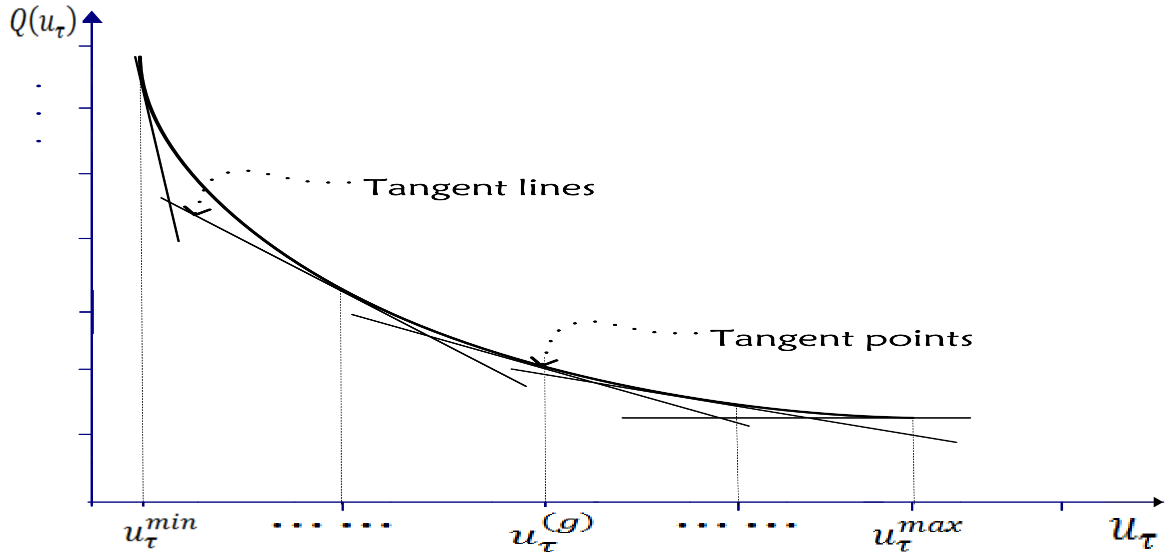


Figure 3: Outer-approximation using tangent lines

To simplify this matter, we use γ tangent lines for approximating all the functions of $Q(u_\tau)$. Give an example in Figure 3, the tangents points are denoted by $u_\tau^{(1)} = u_\tau^{min}$, $u_\tau^{(2)}, u_\tau^{(g)} \cdots, u_\tau^{(\gamma)} = u_\tau^{max}$ for all the $Q(u_\tau)$ functions. These points are evenly distributed in the domain that corresponds to the function value. Given these tangent points, the tangent lines can be formulated accordingly. For the point $u_\tau^{(g)}$ as an example, the tangent line can be calculated as: $y = Q'\left(u_\tau^{(g)}\right) \times \left(u_\tau - u_\tau^{(g)}\right) + Q\left(u_\tau^{(g)}\right)$. By using the tangent lines to approximate the non-linear functions in the objective function (13) can be formulated as follows:

$$Minimize \ \sum_{\tau=1}^{|\mathcal{H}|} \lambda_\tau \times L_\tau \qquad (42)$$

$$\lambda_\tau \geq Q'\left(u_\tau^{(g)}\right) \times \left(u_\tau - u_\tau^{(g)}\right) + Q\left(u_\tau^{(g)}\right) \qquad \tau = 1,2,\cdots,|\mathcal{H}|; \ \ g = 1,2\cdots\gamma$$

(43)

$\lambda_\tau \geq 0 \qquad\qquad \tau = 1,2,\cdots|\mathcal{H}|$

(44)

After above linearization and outer-approximation, the original model ($\boldsymbol{\mathcal{M}}$_$\mathbf{2}$) can be approximated by a mixed-integer linear programming model ($\boldsymbol{\mathcal{M}}$_$\mathbf{3}$) as follows:

($\boldsymbol{\mathcal{M}}$_$\mathbf{3}$) *Minimize* $\sum_{\tau=1}^{|\mathcal{H}|} \lambda_\tau \times L_\tau$

s.t. Constraints (14)−(31), (34)−(35) , (37)− (38),(43) −(44) .

5. Solution approaches

To verify the performance of the proposed model ($\boldsymbol{\mathcal{M}}$_$\mathbf{3}$), the CPLEX solver is used to solve small instances. However, for large-scale problem instances, the model becomes too intractable for the CPLEX to solve directly. This study suggests two solution approaches to solve the proposed model ($\boldsymbol{\mathcal{M}}$_$\mathbf{3}$). One is the local branching based method; the other is the Particle Swarm Optimization (PSO) based method, which can be used to solve large-scale instances. The following sub-sections would introduce the local branching based solution method and the PSO based solution method, respectively.

5.1 Local branching based solution method

From the structure of model ($\boldsymbol{\mathcal{M}}$_$\mathbf{3}$), the model is a mixed integer programming model with two sets of binary variables, i.e. $x_{iq}, z_{ij} \in \{0,1\}$ and two auxiliary binary decision variables E_τ^n, F_τ^n. It is obvious that the core variables of the model are x_{iq}. A local branching algorithm proposed by Fischetti and Lodi (2003) is used to solve this problem. The local branching algorithm aims to find a suboptimal solution and speed up the global search process of a bounding based search scheme. The core idea of the local branching strategy is to use the CPLEX solver to explore suitable solution subspaces, which are defined and controlled at a 'strategic' level by a simple external branching framework. The whole solution method is in the spirit of widely used local search meta-heuristics, and the neighborhoods are formulated by the local branching cuts (Zhen et al., 2016). The main searching procedure of the local branching is illustrated in Figure 4.

In Figure 4, the rectangle denotes the solution space of the problem, i.e., U. Each circle in it denotes a sub-problem which reflects a neighborhood of a feasible solution of the original problem. The main procedure of the sub-problem can be expressed as follows: (1) an initial feasible solution x_1 of the local branching procedure can be obtained by the CPLEX solver; (2) a neighborhood of radius e contains a set of feasible solutions of the MIP model, i.e, x. And from the node x_1 we derive the node x_2' and x_2, which denote the model ($\boldsymbol{\mathcal{M}_3}$) with another constraint $|x - x_1| \leq e$, $|x - x_1| \geq e + 1$, respectively. Here $|x - x_1|$ reflects the radius of

x_1's neighborhood in solution space, $|x - x_1|$is calculated as: $|x - x_1| = \sum_{i=1}^{N} \sum_{q=1}^{Q} |x_{i,q} - x_1|$. The node x_2' denotes the objective is improved and the node will be terminated, the node x_2 is continue to search. (3) The CPLEX solver is used to solve the above sub-problem. When the node x_2 is solved, then the node x_3 can be solved by node x_2, and we then repeat this procedure. It should be mentioned that we impose an upper limit for the computation time in each node. If the computation time exceeds the limit when solving a node, the CPLEX solver will stop solving the node and return a solution, which is feasible but non-optimal for the model. In addition, the radius e (i.e.,$|x - x_1| \leq e$) controls the neighbor size of the searching space, thus a proper value of e should be carefully considered.

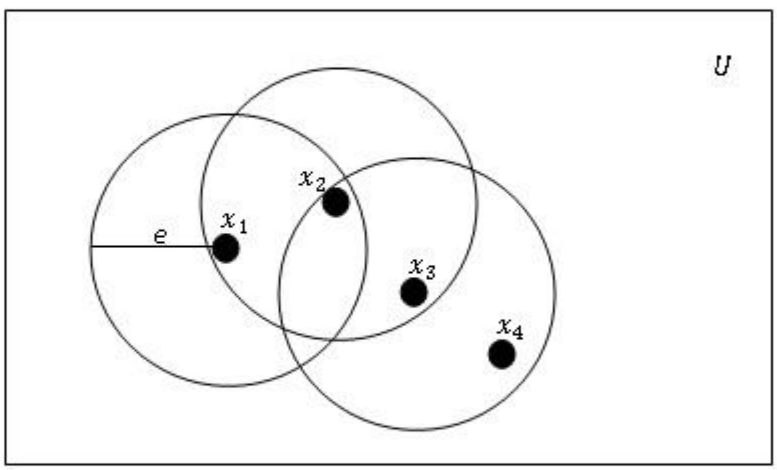
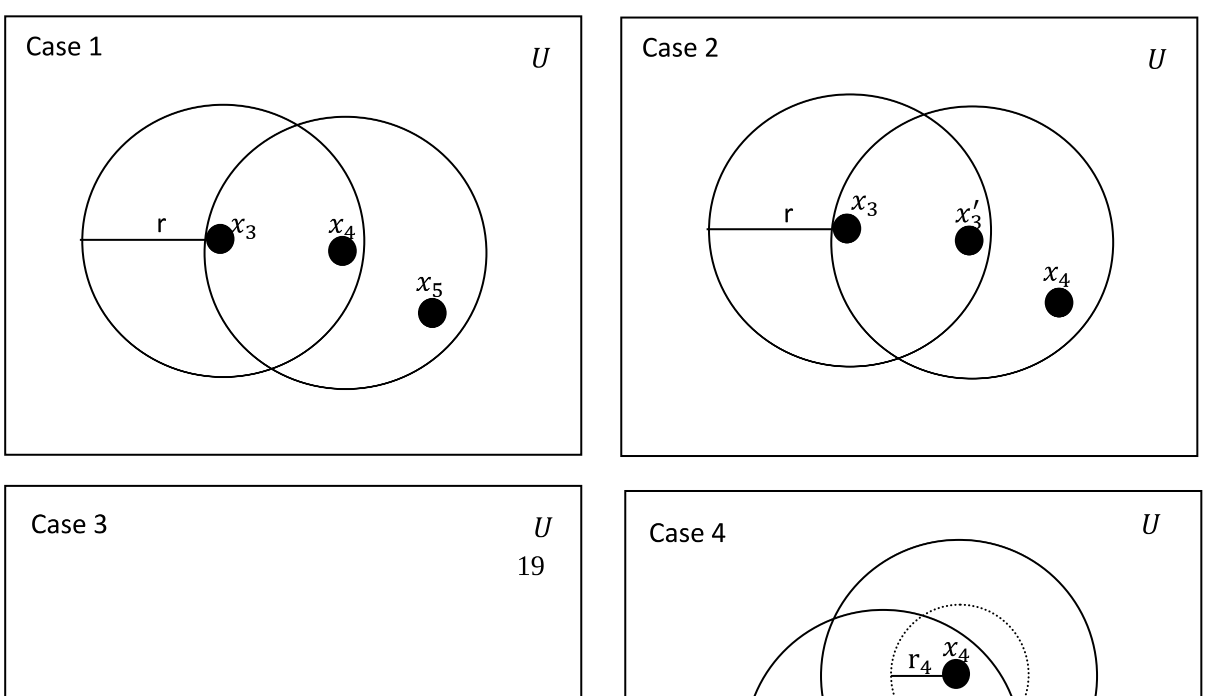


Figure 4: The main searching procedure of local branching

Based on node x_1, there are two nodes created. One is the combination of $\mathcal{M}_3$ and $|x - x_1| \leq e$; the other is the combination of $\mathcal{M}_3$ and $|x - x_1| \geq e + 1$. The circle with the center x_1 in Figure 4 reflects the above first case; here x_2 is the solution in the first case and is inside the circle. While, the above second case is reflected by that x_3 is outside the circle with center x1 and is inside the circle with center x_2; here x_3 is the solution in the second case "$|x - x_2| \leq e$". Figure 4 shows the branching flow is as follows. The first case is solved, and the solution is x_2. Then we solve the second case with "$|x - x_2| \leq e$", and the solution is x_3. Then we solve the second case with "$|x - x_2| \geq e + 1$" and "$|x - x_3| \leq e$", and the solution is x_4.

When solving the sub-problem, the result solved by the CPLEX may have four cases. (Case 1) The objective value of the model is improved, and the solving time limit for the CPLEX solver is not reached. It means the node is better than the incumbent best solution so far. (Case 2) The solving time limit is reached, while the objective value is improved. It indicates the new solution get a new better solution, but it is not an optimal one. (Case 3) The solving time limit is not reached. However, the objective value is not improved. It implies the solution becomes worse, but it is an optimal one for this sub-problem. (Case 4) The solving time limit is reached, and the objective value is not improved. It means the solution is not an optimal solution and is improved the incumbent best solution so far.

The handling strategies of the four cases are shown in Figure 5. For case 1 and case 2, it means that the branching processes evolve the initial solution to a better one, i.e., x_3' or x_4. x_3' or $x_4$becomes the incumbent best one so far. For case 3, the node is completely pruned. In case 4, in order to search a better solution in a faster way, the searching size of the neighborhood is reduced. As shown in the lower right quarter of Figure 5 we can observe that the neighborhood reduced from the size of r_3 to r_4. The above four cases cover all the possible situations in the whole procedure. In this study, the neighborhoods used in local branching procedure are realized by defining some linear inequalities (or branching cuts) with respect to binary variables ($x_{i,q}$). And if the solution is not improved after a certain number of consecutive iterations, the whole searching process will be terminated.



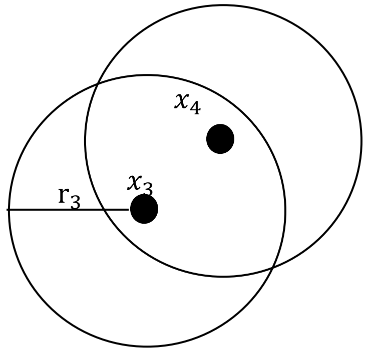


Figure 5: Four cases during the branching process

When an extensive set of numerical experiments carried out to validate the local branching based solution method. We found the method is effective when solving some media medium size instances, while solving the extremely large scale instance is time consuming. Indeed, the local branching solver, as a black-box for performing the tactical branching, requires a starting (feasible) references solution. This solution may requires an excessive computing time especially when the instance size is large (Letchford and Lodi, 2003). So we have to consider some other heuristic solution methods.

5.2 PSO based solution method

Particle Swarm Optimization (PSO) was introduced by Eberhart and Kennedy (1995). Nowadays the PSO algorithm has been widely used for solving port optimization problems, e.g., QC scheduling problem (Wang et al. 2012, Guo et al. 2014), berth allocation problem (Ting et al., 2014). It demonstrates that the PSO algorithm can effectively solve port optimization problems. Thus, this paper designs a PSO based solution approach to solve model ($\boldsymbol{\mathcal{M}\_3}$).

The PSO algorithm is a population-based algorithm, in which each solution is denoted by a particle whose status contains its position and velocity. The position indicates the quality of the solution and the particle's velocity shows the direction it will move in the next iteration. Similar to the local branching based solution strategy, we still use the x_{iq} variable to define particles in the PSO. More specifically, for a particle m in iteration n, its position is defined by $\mathbb{x}_m^n = \{x_{miq}^n\}$, and its velocity is defined by $\mathbb{v}_m^n = \{v_{miq}^n\}$, $\forall i \in \Omega, \forall q \in Q$. The updating formulae of velocity and position are:

$$v_{miq}^{n+1} = w^n \ v_{miq}^n + \ c_1 r_1 \left(XpBest_{miq}^n - X_{miq}^n \right) + c_2 r_2 \left(XgBest_{iq}^n - X_{miq}^n \right) \quad (45)$$

$$X_{miq}^{n+1} = \ X_{miq}^n + v_{miq}^n \quad (46)$$

Where c_1 and c_2 are acceleration weights, r_1 and r_2 are random numbers generated between zero and one. $XpBest_{miq}^n$ is the best position of the particle m until iteration n, $XgBest_{iq}^n$ is the best position of the whole swarm until iteration n. w^n is the inertia weight that calculated by $w^n = \frac{N-n}{N}(w_{in} - w_{en}) + w_{en}$, in which N is the number of iterations, w_{in} and w_{en} are the initial inertia weight and the final inertia weight, respectively. The inertia weight affects the PSO procedure's convergence towards the optimal solution.

The standard PSO algorithm may grow limitless and this will affect the convergence to the best solution. To overcome this problem, Shi and Eberhart (1998) proposed a new velocity updating formula which imposes a particle not only move in the direction of the best positions but may also fly to two randomly chosen particles. Then formula (43) can be modified as follows:

$$v_{miq}^{n+1} = w^n \ v_{miq}^n + \ c_1 r_1 \left(XpBest_{miq}^n - X_{miq}^n \right) + c_2 r_2 \left(XgBest_{iq}^n - X_{miq}^n \right) + c_3 r_3 \left(Xa_{iq}^n - Xb_{iq}^n \right) \quad (47)$$

Here Xa_{iq}^n and Xb_{iq}^n are the positions of two randomly chosen particles in iteration n.

Because the decision variables in this study are binary variables, we cannot calculate a particle's position according to the above Formula (44). It is thus modified as follows:

$$X_{miq}^{n+1} = \begin{cases} 1, & when \quad r < S(v_{miq}^n) \\ 0, & when \quad r \geq S(v_{miq}^n) \end{cases} \quad (48)$$

Here r is a random number generated between zero and one, $S(v_{miq}^n)$ is a sigmoid function, $S\left(v_{miq}^n\right) = \frac{1}{1-e^{-v_{miq}^n}}$. Based on the above definition of the solution representation and velocity updating strategy, the main framework of the PSO procedure is as follows.

Step 1: Set the iteration number $n = 1$. Initialize a swarm containing M particles whose positions determine a group of tasks are assigned to a certain QC during each period. The assignment plan is randomly generated.

Step 2: For certain particles (e.g., particle m), if the 'tasks-QC' assignment plan disobeys Constraints (17) of the model that tasks in the same bay must obey the precedence constraints, then a revision should be performed to make sure particle m's assignment meet the actual constraints.

Step 3: Calculate the fitness value of all the particles by solving model ($\boldsymbol{\mathcal{M}\_3}$).

Step 4: Renew the best position of each particle and the best position of the swarm at each iteration, i.e., $\{XpBest_{miq}^{n}\}$, $\{XgBest_{iq}^{n}\}$.

Step 5: Renovate the velocity and the position according to Formula (45) and Formula (46), respectively.

Step 6: Let $n = n + 1$ and then go to Step 2 if the iteration number not reaches the preset maximum value, otherwise, stop.

6. Numerical experiments

Several experiments are performed to validate the efficiency of the solution method and the effectiveness of the proposed model ($\boldsymbol{\mathcal{M}\_3}$). All experiments are performed by CPLEX 12.6 with technology of C# (VS2012) on a DELL Precision 7600 workstation with two Xeon E5-2643 V3 CPUs (24 cores) of 3.4 GHz processing speed and 128 GB of memory running Windows 7.

Experiments on small-scale instances are conducted first to validate the efficiency of the local branching based method. The optimal results are obtained by using the CPLEX solver to solve the model directly.

Table 1: Comparison between the local branching algorithm and the CPLEX solver

task-QC-ship		CPLEX solving directly		Local branching method		OBJ Gap
Scale	ID	OBJ(Ton)	Time(s)	OBJ(Ton)	Time(s)	
5 tasks	5-2-2-1	1290.62	0.37	1290.62	8.40	0.00%
&	5-2-2-2	1230.67	0.34	1230.62	10.82	0.00%
2 QCs	5-2-2-3	1312.43	0.36	1324.65	10.32	0.93%
&	5-2-2-4	1320.79	0.36	1327.10	12.21	0.48%
2 ships	5-2-2-5	1302.31	0.30	1302.32	11.33	0.00%
10 tasks	10-2-2-1	1409.15	117.80	1447.96	37.23	2.75%
&	10-2-2-2	1384.76	136.26	1403.94	46.93	1.38%
2 QCs	10-2-2-3	1349.84	128.58	1378.91	54.95	2.15%
&	10-2-2-4	1355.46	147.49	1387.62	42.86	2.37%
2 ships	10-2-2-5	1399.31	116.18	1418.45	49.92	1.36%
12 tasks	12-3-2-1	1694.13	390.17	1713.64	127.29	1.15%
&	12-3-2-2	1674.55	206.39	1687.52	162.93	0.07%
3 QCs	12-3-2-3	1738.78	377.21	1769.67	216.49	1.78%
&	12-3-2-4	1644.95	295.88	1681.61	119.47	2.23%

2 ships	12-3-2-5	1697.87	271.04	1740.46	138.05	2.51%
15 tasks	15-3-2-1	1754.83	2344.98	1772.47	260.41	1.01%
&	15-3-2-2	1634.04	2998.39	1659.56	217.92	1.56%
3 QCs	15-3-2-3	1872.92	2763.91	1900.43	432.95	1.47%
&	15-3-2-4	1843.09	1976.28	1894.31	484.78	2.78%
2 ships	15-3-2-5	1824.19	2883.15	1878.29	542.84	2.96%
Avg.			757.77		139.41	1.45%

Notes: (1) OBJ Gap= (OBJ (Local branching) – OBJ (CPLEX))/ OBJ (CPLEX)

Table 1 shows that the local branching based method can obtain near-optimal results. The average gap between the objectives of the local branching based method and the optimal results is 1.45%. The CPU time for the local branching based method is longer than the CPLEX solver's time when solving some small-scale problem cases, but is much shorter than the CPLEX solver when the problem size is large. With the size of the problem increasing, the CPLEX solver's computation time grows very rapidly, near exponential rate growth; while the CPU time of the local branching based method also grows, but the growth rate is not very fast. The CPLEX solver cannot solve the model directly under large-scale problem cases, whereas the local branching based method can solve such problems within a reasonable time period. Besides the local branching based solution method, another method based on the PSO is also used to solve large-scale problems. The performance of the proposed two solution methods is also investigated by comparative experiments. The results are shown in Table 2 as follows:

Table 2: Comparison between the local branching algorithm and the PSO method

task-QC-ship		**Local branching method**		**PSO method**		OBJ Gap
Scale	ID	OBJ(Ton)	Time(s)	OBJ(Ton)	Time(s)	
20 tasks	20-2-3-1	2810.26	664.17	2898.41	844.41	3.15%
&	20-2-3-2	2914.60	480.65	3017.25	916.74	3.52%
2 QCs	20-2-3-3	2636.45	658.56	2668.18	908.96	1.20%
&	20-2-3-4	2637.56	732.97	2710.45	880.38	2.76%
3 ships	20-2-3-5	2631.74	613.52	2671.75	975.67	1.52%
30 tasks	30-3-3-1	3062.31	3150.88	3095.09	1618.31	1.07%
&	30-3-3-2	3109.40	3602.52	3163.26	1524.35	1.73%
3 QCs	30-3-3-3	2885.93	3759.43	2927.71	1729.21	1.45%
&	30-3-3-4	2834.12	3606.74	2914.35	1528.22	2.83%
3 ships	30-3-3-5	2911.63	3394.27	2985.86	1775.83	2.55%

35 tasks	35-4-4-1	3410.24	5863.10	3487.76	2928.09	2.27%
&	35-4-4-2	3381.82	5677.94	3403.16	3195.81	0.63%
4 QCs	35-4-4-3	3353.42	6290.81	3386.25	2744.23	0.98%
&	35-4-4-4	3495.49	5337.62	3553.16	2783.32	1.64%
4 ships	35-4-4-5	3268.14	5780.26	3314.27	3203.16	1.41%
40 tasks	40-4-4-1	N.A.	>9999.00	3890.17	4425.82	N.A.
&	40-4-4-2	N.A.	>9999.00	3785.94	4489.76	N.A.
4 QCs	40-4-4-3	N.A.	>9999.00	3834.33	4864.81	N.A.
&	40-4-4-4	N.A.	>9999.00	3792.91	4178.27	N.A.
4 ships	40-4-4-5	N.A.	>9999.00	3953.58	4297.23	N.A.
Avg.						1.91%

Notes: (1) OBJ Gap= (OBJ (PSO) – OBJ (Local branching))/ OBJ (Local branching)

As shown in Table 2, we can observe that the PSO based method can obtain near-optimal solutions according to the comparison with the local branching based method. The results show that the average gap between the results of the PSO based method and the local branching based method is less than 1.91%. The CPU time of the local branching based method increases evidently when the problem size increases. For some extremely large-scale problems, the local branching based method cannot solve them within a reasonable time, but the PSO based method can solve them. The results indicate that the PSO based method has its relative merit, and can obtain satisfying solutions in a much faster way than the local branching based method for some extremely large-scale problems.

Based on the above comparison experiments, we summarize the results in Table 3.

Table 3: Comparison among the three solution methods

Instances		CPLEX solving directly		**Local branching method**		**PSO method**	
Scale	**ID**	OBJ(Ton)	Time(s)	OBJ(Ton)	Time(s)	OBJ(Ton)	Time(s)
5 tasks	5-2-2-1	1290.62	0.37	1290.62	8.40	1293.43	69.23
2 QCs	5-2-2-2	1230.67	0.34	1230.62	10.82	1235.84	56.06
2 ships	5-2-2-3	1312.43	0.36	1324.65	10.32	1315.46	52.37
10tasks	10-2-2-1	1409.15	117.80	1447.96	37.23	1453.21	97.92
2 QCs	10-2-2-2	1384.76	136.26	1403.94	46.93	1405.07	81.25
2ships	10-2-2-3	1349.81	128.58	1378.91	54.95	1396.91	79.28
15tasks	15-3-2-1	1754.83	2344.98	1772.47	260.41	1774.37	345.19

3 QCs	15-3-2-2	1634.04	2998.39	1659.56	217.92	1695.87	353.24
2 ships	15-3-2-3	1872.92	2763.91	1900.43	432.95	1907.63	376.27
20tasks	20-2-3-1	2773.17	5439.47	2810.26	664.17	2898.41	844.41
2 QCs	20-2-3-2	2863.30	5875.61	2914.60	480.65	3017.25	916.74
3 ships	20-2-3-3	2610.61	6099.04	2636.45	658.56	2668.18	908.96
30tasks	30-3-3-1	N.A.	>7500.00	3062.31	3150.88	3095.09	1618.31
3 QCs	30-3-3-2	N.A.	>7500.00	3109.40	3602.52	3163.26	1524.35
3 ships	30-3-3-3	N.A.	>7500.00	2885.93	3759.43	2927.71	1729.21
40tasks	40-4-4-1	N.A.	>9999.00	N.A.	>9999.00	3890.17	4425.82
4 QCs	40-4-4-2	N.A.	>9999.00	N.A.	>9999.00	3785.94	4489.76
4 ships	40-4-4-3	N.A.	>9999.00	N.A.	>9999.00	3834.33	4864.81
50tasks	50-6-6-1	N.A.	>9999.00	N.A.	>9999.00	5625.72	6317.28
6 QCs	50-6-6-2	N.A.	>9999.00	N.A.	>9999.00	5350.06	5980.21
6 ships	50-6-6-3	N.A.	>9999.00	N.A.	>9999.00	5669.29	5762.43
Avg.							1947.29

We recommend that port schedule planners could use the CPLEX solver to solve small-scale problems, the local branching based solution method to solve middle-scale problems, and the PSO based solution method to solve large-scale problems.

7. Conclusions

This study considers the factors of tide and fuel consumption in the traditional QCSP. We propose a nonlinear mixed-integer programming (NMIP) model and linearize the nonlinear parts in the model by some approximation approaches. The local branching based solution method as well as the PSO based solution method is suggested to solve the proposed model in large-scale problem instances. Numerical experiments with some real-world like cases are conducted to validate the effectiveness of the proposed model and the efficiency of the proposed solution methods. The proposed methodology in this study may be potentially useful for some tidal ports (e.g., Shanghai Port) where mega-ships need to take advantage of the tide so as to pass through the navigation channels in the ports. Moreover, the QC scheduling plan made by the proposed model does not only benefit the port operator but also considers the fuel consumption factor that concerns the shipping liners.

The main contributions of this study mainly have three aspects: (1) it extends the traditional QCSP by considering the tidal factors and fuel consumptions for the served ships. (2) A nonlinear MIP model is proposed and linearized so as to become solvable. (3) Two types of solution methods are designed to solve the new model with large-scale instances during a reasonable time.

However, this study also contains some limitations. For example, the time for ships passing through the channel is assumed to be identical; the capacity limitation of navigation channel is not considered. QC scheduling may also influence the traffic congestion in yard (Zhen, 2016). All of these issues will be our research directions in the future.

Acknowledgements

The authors would like to thank anonymous reviewers for their valuable comments and constructive suggestions, which have greatly improved the quality of this paper. This research is supported by the National Natural Science Foundation of China (71422007), Shanghai Social Science Research Program (2014BGL006).

References

Bierwirth, C., & Meisel, F. (2009). A fast heuristic for quay crane scheduling with interference constraints. *Journal of Scheduling*, 12(4), 345-360.

Bierwirth, C., & Meisel, F. (2010). A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3), 615-627.

Chen, J. H., Lee, D. H., & Goh, M. (2014). An effective mathematical formulation for the unidirectional cluster-based quay crane scheduling problem. *European Journal of Operational Research*, 232(1), 198-208.

Daganzo, C. F. (1989). The crane scheduling problem. *Transportation Research Part B*, 23(3), 159-175.

Du, Y., Chen, Q., Quan, X., Long, L., & Fung, R. Y. (2011). Berth allocation considering fuel consumption and vessel emissions. *Transportation Research Part E*, 47(6), 1021-1037.

Du, Y., Chen, Q., Lam, J. S. L., Xu, Y., & Cao, J. X. (2015). Modeling the Impacts of Tides and the

Virtual Arrival Policy in Berth Allocation. *Transportation Science*, 49(4), 939 – 956.

Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science 1*, 39-43.

Fagerholt, K. (2004). A computer-based decision support system for vessel fleet scheduling experience and future research. *Decision Support System*s, 37(1), 35-47.

Golias, M. M., Saharidis, G. K., Boile, M., Theofanis, S., & Ierapetritou, M. G. (2009). The berth allocation problem: Optimizing vessel arrival time. *Maritime Economics & Logistics*, 11(4), 358-377.

Guo, P., Cheng, W., & Wang, Y. (2014). A modified generalized extremal optimization algorithm for the quay crane scheduling problem with interference constraints. *Engineering Optimization*, 46(10), 1411-1429.

He, J. L., Huang, Y. F. and Chang, D. F. (2015). Simulation-based heuristic method for container supply chain network optimization. *Advanced Engineering Informatics*, 29(3), 339-354.

Hu, Q. M., Hu, Z. H., & Du, Y. (2014). Berth and quay-crane allocation problem considering fuel consumption and emissions from vessels. *Computers & Industrial Engineering*, 70, 1-10.

Kim, K. H., & Park, Y. M. (2004). A crane scheduling method for port container terminals. *European Journal of Operational Research*, 156(3), 752-768.

Lee, D. H., Wang, H. Q., & Miao, L. (2008). Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E*, 44(1), 124-135.

Legato, P., Trunfio, R., & Meisel, F. (2012). Modeling and solving rich quay crane scheduling problems. *Computers & Operations Research*, 39(9), 2063-2078.

Letchford, A. N., & Lodi, A. (2003). An augment-and-branch-and-cut framework for mixed 0-1 programming. *In Combinatorial Optimization—Eureka, pp*. 119-133. *Springer Berlin Heidelberg*.

Lim, A., Rodrigues, B., & Xu, Z. (2007). A m-parallel crane scheduling problem with a non-crossing constraint. *Naval Research Logistics*, 54(2), 115-127.

Liu, J., Wan, Y. W., & Wang, L. (2006). Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures. *Naval Research Logistics*, 53(1), 60-74.

MAN Diesel & Turbo, 2004. Basic Principles of Ship Propulsion. <http://www.manbw.com/files /news/filesof3859/P254-04-04.pdf>. (accessed 2015.09.10).

Meisel, F., & Bierwirth, C. (2011). A unified approach for the evaluation of quay crane scheduling models and algorithms. *Computers & Operations Research*, 38(3), 683-693.

Meng, Q., Wang, S., Andersson, H., & Thun, K. (2013). Containership routing and scheduling in liner shipping: overview and future research directions. *Transportation Science*, 48(2), 265-280.

Moccia, L., Cordeau, J. F., Gaudioso, M., & Laporte, G. (2006). A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics*, 53(1), 45-59.

Ng, W. C., & Mak, K. L. (2006). Quay crane scheduling in container terminals. *Engineering Optimization*, 38(6), 723-737.

Notteboom, T. E. (2006). The time factor in liner shipping services. *Maritime Economics & Logistics*, 8(1), 19-39.

Peterkofsky, R. I., & Daganzo, C. F. (1990). A branch and bound solution method for the crane scheduling problem. *Transportation Research Part B*, 24(3), 159-172.

Port of Antwerp (2016), Accessed Aprial 17, 2016. https://www.hafen-hamburg.de/en

Qureshi, A. G., Taniguchi, E., & Yamada, T. (2009). An exact solution approach for vehicle routing and scheduling problems with soft time windows. *Transportation Research Part E*, 45(6), 960-977.

Rakke, J. G., Christiansen, M., Fagerholt, K., & Laporte, G. (2012). The traveling salesman problem with draft limits. *Computers & Operations Research*, 39(9), 2161-2167.

Ronen, D. (1982). The effect of oil price on the optimal speed of ships. *Journal of the Operational Research Society*, 1035-1040.

Ronen, D. (2011). The effect of oil price on containership speed and fleet size. *Journal of the Operational Research Society*, 62(1), 211-216.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *IEEE World Congress on Computational Intelligence* 69-73.

Song, J. H., & Furman, K. C. (2013). A maritime inventory routing problem: Practical approach. *Computers & Operations Research*, 40(3), 657-665.

SHMSA（2016）Accessed April 16, 2016, http:// www.Shmsa.gov.cn.

Tavakkoli-Moghaddam, R., Makui, A., Salahi, S., Bazzazi, M., & Taheri, F. (2009). 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.

Tawarmalani, M., & Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99(3), 563-591.

Ting, C. J., Wu, K. C., & Chou, H. (2014). Particle swarm optimization algorithm for the berth allocation problem. *Expert Systems with Applications*, 41(4), 1543-1550.

Wang, S., & Meng, Q. (2011). Schedule design and container routing in liner shipping. *Transportation Research Record*, (2222), 25-33.

Wang, S., & Meng, Q. (2012). Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E*, 48(3), 701-714.

Wang, S., Zheng, J., Zheng, K., Guo, J., & Liu, X. (2012). Multi resource scheduling problem based on an improved discrete particle swarm optimization. *Physics Procedia*, 25, 576-582.

Xu, D., Li, C. L., & Leung, J. Y. T. (2012). Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*, 216(1), 47-56.

Zhu, Y., & Lim, A. (2006). Crane scheduling with non-crossing constraint. *Journal of the Operational Research Society*, 57(12), 1464-1471.

Zhen, L. (2015) Tactical berth allocation under uncertainty. *European Journal of Operational Research* 247: 928-944.

Zhen, L., Xu, Z., Wang, K., & Ding, Y. (2016). Multi-period yard template planning in container terminals. *Transportation Research Part B*, in press. doi:10.1016/j.trb.2015.12.006.

Zhen, L. (2016) Modeling of yard congestion and optimization of yard template in container ports. *Transportation Research Part B* 90: 83-104.

Liu, Z., Wang, S., Chen, W., Zheng, Y., 2016. Willingness to board: A novel concept for modeling queuing up passengers. Transportation Research Part B, Vol. 90, pp. 70–82.