The following publication Wu, L., Yang, D., Wang, S., & Yuan, Y. (2020). Evacuating offshore working barges from a land reclamation site in storm emergencies. Transportation Research Part E: Logistics and Transportation Review, 137, 101902 is available at https://doi.org/10.1016/j.tre.2020.101902.

Evacuating offshore working barges from a land reclamation site in storm emergencies

Abstract

This paper investigates a barge evacuation planning problem (BEPP) that can arise during land reclamation projects. The problem was motivated by the issue faced in actual practice by the Hong Kong International Airport (HKIA). In this problem, a fleet of heterogeneous barges working at an offshore land reclamation site needs to be evacuated to coastal shelters prior to the arrival of a storm. Having no propulsion power of their own, these barges must be towed by tug boats in order to be evacuated. The problem under consideration is very complicated since it involves a series of inter-correlated assignment and scheduling decisions at different planning levels. To solve the problem, this paper first formulates the problem as a nonlinear Mixed Integer Programming (MIP) model. The model is then linearized. We further proved that the BEPP is NP-hard in the strong sense. In view of the complexity of the problem, a tailored heuristic method is proposed. Extensive numerical experiments and a case study are performed, and the results demonstrate the effectiveness and efficiency of the solution method. Land reclamation projects have become increasingly popular in recent years, and the proposed method is applicable to solving BEPPs arising in similar scenarios.

Keywords: Land Reclamation, Disaster Prevention, Barge Evacuation, Heuristic

1. Introduction

As one potential solution to satisfying the increasing demand for new land for living and development, offshore land reclamation has become increasingly popular in coastal areas around the globe (Martín-Antón et al., 2016). In the past, many coastal countries, such as China (Wang et al., 2014), Japan (Suzuki, 2003), South Korea (Son and Wang, 2009), Singapore (Glaser et al., 1991), the Netherlands (Hoeksema, 2007), and the UK (OSPAR Commission, 2008), have conducted massive land reclamation projects for coastal city expansion, both for land space for industrial and agricultural development in coastal areas, and for defense against storm surges.

Various types of working vessels are needed for a land reclamation project. Figure 1 shows the land reclamation project for the Hong Kong International Airport (HKIA). It is noticed that most of these vessels are non-self-propelled working barges that rely on the assistance of tug boats to move. In land reclamation projects, the most commonly used barge types include pontoons, floating cranes, sanding barges, flat bottom barges, and pump dredgers. Barges used in a land reclamation project site are usually large in size; for example, a floating crane can be nearly 100 meters long and weigh more than 20,000 tons.

Barges are extremely vulnerable to severe weather conditions (e.g., storms at sea). Therefore, to ensure safety, working barges must be evacuated from a land reclamation project site to shelters (i.e., coastal harbors) before storms arrive. Barge evacuation is a very complicated problem. This is because (i) the number of working barges can be very large (a medium-scaled reclamation project may have more than 50 barges working simultaneously), (ii) the evacuation time window can be narrow (all barges need to be evacuated within no more than 2 or 3 days, which is basically the

Figure 1: Barges working for a land reclamation project (source: HKIA).

time interval between the forecast and arrival of a storm), and (iii) evacuation involves careful coordination among the barges and various limited resources (e.g., tug boats, site channels, and shelters).

As one of the world's busiest airports, HKIA is currently conducting a huge land reclamation project for constructing its third runway system (3RS) in order to meet its ever-growing air traffic demand (Hong Kong International Airport, 2018). Hong Kong is located in a region that frequently suffers from typhoons during the summer period from June to October. When a typhoon is forecast, the barges must be evacuated from the 3RS project site to a group of shelters prior to the arrival of the typhoon.

The Barge Evacuation Planning Problem (BEPP) that arises for HKIA has a complex structure and involves decisions at different planning levels. To be more specific, at the strategic level the construction party (i.e., HKIA) must decide the assignment of shelters to barges. Following this, the assignment of tug boats among the various procedures of the evacuation must be determined at the tactical level (the evacuation of barges consists of three procedures, including in-site tugging, open-sea tugging, and in-shelter tugging). Finally, decisions regarding the times at which to start each procedure for evacuating each barge, as well as the tug boats assigned to serve each barge in each procedure, must be made at the operational level. Note that these decisions are interconnected, which makes the problem even harder. To better present the problem, we will formulate it as a Mixed Integer Programming (MIP) model. We also demonstrate that the problem is NP-hard in the strong sense. In view of its complexity, a heuristic algorithm, which takes advantage of special features of the problem, is proposed to solve the problem efficiently.

Evacuation is a hot topic in transportation studies, and most studies in this area focus on the evacuation of residents under emergency situations. The major topics include (i) behavior modeling of evacuees (e.g., Ng et al., 2015 and Fry and Binner, 2016), (ii) evacuation network planning for residents (e.g., Stepanov and Smith, 2009 and Xie et al., 2010), and (iii) evacuation planning and control (e.g., Yi et al., 2017 and Karabuk and Manzour, 2019). For a comprehensive understanding of researches on evacuation planning and management of residents, refer to the recent surveys of Murray-Tuite and Wolshon (2013) and Bayram (2016).

It is noticed that the evacuation of vessels under emergency situations has attracted limited attention compared with the rich literature on the evacuation of residents and, to the best of our knowledge, there are no existing studies that focus on the evacuation of non-self-propelled barges. Some studies focus on the evacuation of self-propelled vessels under emergencies. One of such studies was conducted by Pitana and Kobayashi (2009). In their work, an evacuation problem was considered for vessels in Osaka Bay in Japan that were threatened by a tsunami. They applied a simulation-based method to optimize the evacuation sequence among the vessels. In another study, Zhao et al. (2017) considered a fishing boat evacuation problem during typhoon emergencies. The problem they studied is by nature an assignment problem between a set of fishing boats and a set of harbors, and to solve the problem efficiently they proposed a simulated annealing heuristic.

Each year, there are various land reclamation projects underway globally, and most of them may face similar barge evacuation problems, just like HKIA. For example, in October 2018 the Hong Kong government announced a huge land reclamation project for the construction of an artificial island with a total area of about 1700 hectares (Hong Kong Government, 2018). On top of that, a number of land reclamation projects are currently underway or going to be conducted in Japan (Japan Property Central, 2018) and Monaco (Scott, 2018), where storms frequently visit. Hence, our method which is proposed to solve HKIA's case can provide a reference for solving the BEPPs arising in other land reclamation projects.

The remainder of this paper is organized as follows. Section 2 provides a detailed description of the considered problem. In Section 3, we formulate an MIP model for the problem, discuss its complexity, and present the solution procedure. To test the performance of the algorithm, we conduct extensive numerical experiments and a case study in Section 4. Our findings and the managerial insights are discussed in Section 5. Finally, we present the conclusions and future research directions in Section 6.

2. The Barge Evacuation Planning Problem

Currently, a fleet of working barges is hired by HKIA (i.e., the construction party) for a land reclamation project in an offshore area where the 3RS is being built. As shown in Figure 2, when an approaching storm is forecast, these barges must be evacuated from the project site to a group of shelters prior to the deadline (i.e., the time when the storm arrives).

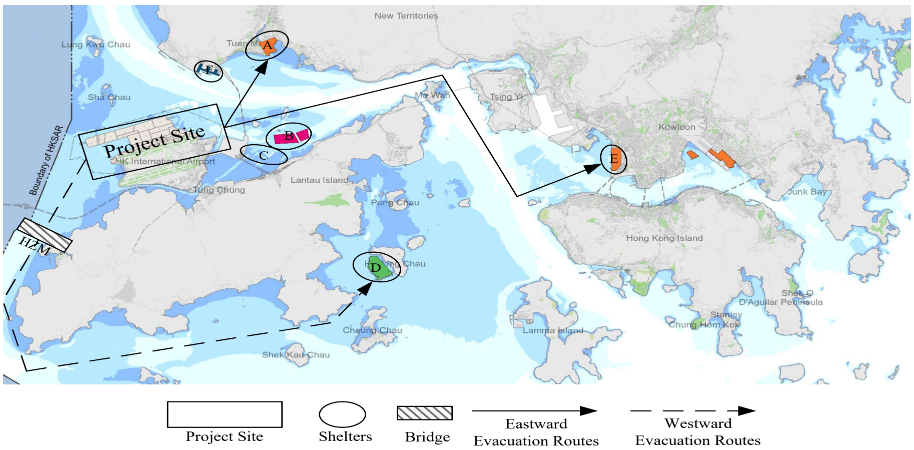

Figure 2: An overview of the barge evacuation (source: HKIA).

The BEPP is to assign a set of resources (i.e., tug boats, site channels, and shelter slots) and schedule the evacuation procedures (i.e, leaving the site, sailing in the open sea, and berthing into the shelter) among barges in order to evacuate all of them to shelters before the deadline. Considering that the cost of renting a barge is very high (e.g. for a medium size floating crane, the daily rent cost is up to 100,000 US dollars), it is crucial to postpone the evacuation start time for as long as possible so as to reduce the non-working hours of the barges.

In the next part of this section, we will first introduce the necessary resources in Section 2.1, and then the procedures of the evacuation are explained in Section 2.2. The resource assignment is illustrated in Section 2.3 and the decisions and the objective for the whole problem are provided in Section 2.4.

2.1. Barges and Evacuation-supporting Resources

Barges used in the project are non-self-propelled vessels with special engineering functions. Tug boats are necessary when a barge moves from one site to another. The barges are heterogeneous in terms of (i) length, (ii) draft, and (iii) height. Figure 3 demonstrates an example of a tugging operation involving a loading crane barge.

Figure 3: An illustration for tugging operations (Dana, 2009).

The resources used in an evacuation include tug boats, site channels, and shelters. Tug boats are used to assist the barges to evacuate. They can be generally classified into two categories, according to their Bollard Pulls (BP). The BP of Light Tugs (LT) is less than 100 tons and can only be used for tugging operations within the project site and the shelters; the BP of Heavy Tugs (HT) is no less than 100 tons and can be used for tugging barges both across the open sea and in the project site or the shelters. Note that the number of tug boats needed to move barges varies according to different evacuation procedures, and may also vary for barges of different size (i.e., larger barges typically require more tug boats). Figure 4 shows the layout of the project site. It is noted that only three channels allow barges to enter or leave the site, and that all of these are narrow waterways that link the project site with the open sea.

When barges reach the shelters, they also need to be berthed at certain berthages or moored using buoys prior to the arrival of the storm, and then stay in the shelters until the storm has passed. A shelter has capacity limitations, these being in two dimensions, including (i) the total number of barges it can accommodate (especially when mooring buoys are needed for berthing barges in the shelter) and (ii) the total space it has to accommodate barges. In addition, the size of barges that the shelters can accommodate may also vary (i.e., each shelter has its own regulations as to the largest size of a barge that it can harbor). In general, there are two types of shelters. The first type of shelter (which is also the most commonly used type) is a vessel harbor with a breakwater built around it. The second type of shelter has no breakwater, so vessels have to moor in such shelters using buoys. We refer to the first and the second type of shelters as Type I shelters and Type II shelters, respectively.

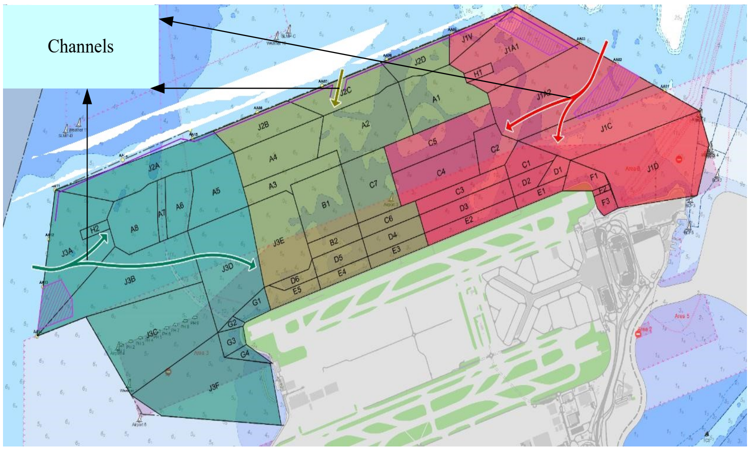

Figure 4: Layout of the project site (source: HKIA).

2.2. *Evacuation Procedures*

The evacuation of a barge is composed of three procedures, these being, in a chronical order, in-site tugging, open-sea tugging, and in-shelter tugging. Figure 5 demonstrates the procedures involved in the evacuation, where the solid line indicates the movement of barges and the dotted lines show the movements of tug boats assigned to serve in the different procedures. In this section, we elaborate on each evacuation procedure, and highlight the practical constraints and considerations.

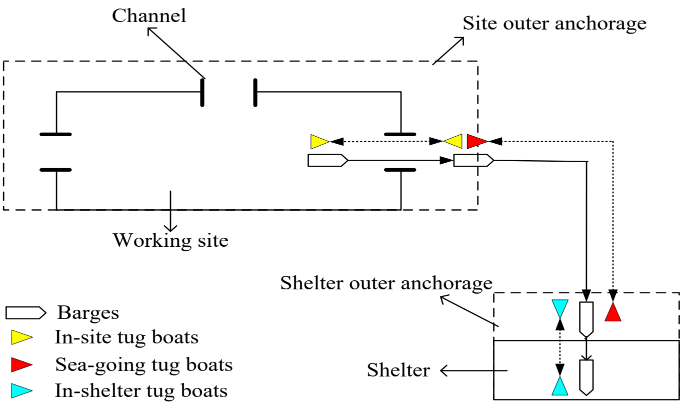

Figure 5: An illustration of the evacuation procedures.

2.2.1. *In-site Tugging*

In this first procedure, a barge is tugged out of the project site. The procedure starts when a sufficient number of tug boats (both HTs and LTs are usable) reach the barge, which will then be tugged to join a departure queue of a channel. The procedure is completed when the barge is tugged out of the project site. After that, the barge will be moored in the outer anchorage of the site, waiting to be tugged across the open sea, and the tug boats used in this first procedure will sail back into the site for subsequent in-site tugging operations.

When barges are tugged into a channel, due to the limited widths of the channels and to avoid congestions, they have to queue up in the channel waiting to depart from the site. In practice, many waterways require barges (ships) to queue up to pass through them. For example, ships have to queue up in the navigational channel of a port when entering or leaving the port (Corry and Bierwirth, 2019; Li and Jia, 2019). Besides, for safety considerations, a minimum time headway must be ensured between two barges that sail consecutively along the same channel. In addition, precedence constraints caused by blockages may exist between two barges that sail along the same channel. For example, a barge may be located near the entrance of a certain channel, so that it has to be tugged out of the site before other barges, so as to leave the lane open for other barges. Finally, tidal conditions can lead to changes in water depth in these channels, hence, barges with large drafts can only sail through such channels during periods of high tide.

2.2.2. Open-sea Tugging

After sailing out of the project site, barges will be tugged by tug boats (HTs only) from the outer anchorage of the site towards their designated shelters. It should be noted that the route chosen for open-sea tugging of a barge may be affected by its height. In HKIA's example, as shown in Figure 2, there are two types of routes between the site and the shelters (i.e., eastward routes and westward routes). While all barges can sail along the eastward routes, only barges with a height less than 41 meters can follow the westward routes, due to the limited bridge height. As a result, for each barge, the distance between the site and its designated shelter is determined not only by the location of the shelter but also by the barge's height. After the barge arrives at its shelter, it will first be moored at the outer anchorage of the shelter, waiting to get berthed into the shelter. Meanwhile, the HTs used for tugging the barge sail back to the outer anchorage of the project site for further arrangements.

2.2.3. In-shelter Tugging

The last procedure of the evacuation is in-shelter tugging, which aims to get barges berthed into shelters. Both HTs and LTs can be used in the berthing operation. Note that in-shelter tugging is only required for the Type I shelters (i.e., shelters with a breakwater). Shelters of this type are designed to accommodate a relatively large number of vessels, and each vessel has to be berthed into a designated berthing area. Such shelters can be very crowded during storm emergencies, and to get berthed, barges have to be tugged to their designated berthing areas at a very low speed by following the narrow paths formed by other vessels already berthed in the shelter. After completing each in-shelter tugging task, the tug boats used for this purpose sail back to the outer anchorage of the shelter to serve barges arriving subsequently. Note that congestions may happen in a shelter such that a path of a barge is blocked by another large barge that is turning in the path to get into its berthing area. But such a scenario can be avoided by better allocation of berthing areas and rarely happens in practice. Therefore, we do not consider congestions in shelters. In addition, one can easily modify the model (refer to Section 3.1) to further control the congestion in a shelter. In particular, similar to the minimum headway required between two barges that sail into the same channel in the in-siting tugging procedure, one can add a similar constraint into the model to set a minimum headway between two barges that sail into the same shelter.

In comparison, such in-shelter tugging is not needed for Type II shelters, around which there is no breakwater built. Barges tugged by HTs from the open sea can be tugged into the shelters directly using the same HTs. Besides, the berthing time of barges (i.e., the time taken to tug barges from the outside of the shelter to the mooring anchorages inside the shelters) in such shelters is very short, since they are moored using buoys without designated berthing areas, resulting in barges being sparsely distributed in the shelters (usually no more than 5 vessels can berth in one such shelter).

2.3. Resource Assignment and Scheduling

The BEPP involves resource assignment and scheduling decisions at three different levels. At the strategic level, the construction party should determine how to assign barges to the shelters. Based on the results of shelter assignment, the construction party should then determine, at a tactical level, the numbers of HTs and LTs that will be assigned for in-site tugging and for in-shelter tugging at each Type I shelter, as well as the number of HTs that will be used for open-sea tugging. Finally, at an operational level, decisions are made to schedule the usages of channels during the in-site tugging procedure, and to assign the tug boats among the barges in all evacuation procedures.

2.4. Decisions and the Objective

There are three types of decisions the construction party must make, with detailed descriptions provided as follows:

- At the beginning of the evacuation, the construction party assigns each barge to a certain shelter and decides on the assignments of tug boats among the various evacuation procedures and shelters.
- Then, for each barge, the construction party decides on the start times of each procedure in its evacuation.
- Finally, for each evacuation procedure of each barge, the construction party should decide on the channel (in the in-site tugging procedure) and the tug boat(s) that are used to move the barge.

The objective of the construction party is to reduce the impact of the storm and achieve greater efficiencies of these working barges, whenever the safety of the barges is ensured.

3. The Model, Complexity, and Solution Procedure

In this section, we first present the BEPP as an MIP model. We then show that the problem is NP-hard in the strong sense. Finally, we describe the procedure to solve the problem at the end of the section.

3.1. The Mathematical Model

We formulate a time-discretization MIP model for the BEPP. To do so, we divided the whole evacuation period into a set of discrete time point t's. The evacuation period starts at the time when tug boats have been assigned properly, after an approaching storm is forecast and ends at the deadline of the evacuation (i.e., the expected arrival time of the storm). We use time point $t = 0$ and $t = D$, $D > 0$ to denote the earliest start time and the deadline of the evacuation, respectively.

For scheduling problems like the BEPP, time-discretization models typically have fewer variables and fewer constraints when compared with arc-flow based models. Besides, with a time-discretization model, most constraints can be written as cover inequalities. This will further improve the efficiencies of off-shelf solvers for solving the model, since most solvers work within a branch-and-cut framework.

In the following part of this section, we first identify the assumptions made for the model and introduce the notation used in the model. We then discuss the objective and explain the constraints of the model which is initially presented in a nonlinear form. Finally, the model is linearized to facilitate the better usage of off-shelf solvers.

3.1.1. Model Assumptions

To better analyze the problem, we made the following assumptions based on the practice of HKIA:

A1. All HTs are identical and all LTs are identical.

A2. The assignment of tug boats to evacuation procedures has to be determined at a strategic level before the beginning of the evacuation and remains unchanged during the whole evacuation.

A3. All decisions are made at time point t's within the evacuation period.

Several aspects of these assumptions are worth mentioning. First, Assumption **A1** is made based on the fact that in most scenarios all HTs (resp. LTs) are treated identically when they are used to tug a barge and that they have similar speeds. Second, Assumption **A2** is used to simplify our analysis and it is also in accordance with the practice of HKIA. Finally, the last assumption is a common assumption used in time-discretization formulations.

3.1.2. Notation

The notation for describing the model is listed in Table 1.

Table 1: Notation.

Indices:	
i, j	Indices for barges, arranging in the alphabetical order. i and j can be used to represent any barge, but they represent different barges when used in one constraint. For example, in Constraint (15), the precedence relationship is specified between two barges.
h	Index for types of tug boats, with $h = 1$ and $h = 2$ representing HTs and LTs, respectively.
k	Index for channels of the project site, arranging in the alphabetical order.
g	Index for shelters, arranging in the alphabetical order.
t, t_1	Indices for time points, arranging in the chronological order. These indices can be used to represent any time point. However, when they are used in one constraint (i.e., Constraint (13), (14), (18), or (21)) they represent different sets of time points. See the detailed discussions on Constraints (13) and (14) in Section 3.1.3.
Sets:	
T	Set of all time points in the evacuation period. $T = \{0, 1, ..., D\}$, where D is time point that corresponds to the deadline of the evacuation.
Ω	Set of all barges to be evacuated, $\Omega = \{1, 2, ..., \|\Omega\|\}$.
Φ	Set of types of tug boats, $\Phi = \{1, 2\}$.
Ψ	Set of all channels connecting the project site and the open sea, $\Psi = \{1, 2, ..., \|\Psi\|\}$.
Θ	Set of all shelters, $\Theta = \{1, 2, ..., \|\Theta\|\}$.
Δ_{ki}	Set of time points with suitable tidal conditions for barge i to start sailing in channel k. As we discussed in Section 2.2.1, for barges with large drafts, they can only sail in the channels when the water depth in these channels is sufficient. For example, suppose T contains ten time points and $T = \{1, 2, ..., 10\}$. Suppose for the safety of its hull, barge $i = 1$ can only sail in the channels with water depth of at least four meters. In addition, if the barge sails out of the project site from channel $k = 1$, the sailing time in this channel is two unit times. The water depth in channel 1 is greater than or equal to four meters at time points 4 to 9. Then barge 1 can safely sail into channel 1 at time points 4 to 7. Thus, we have $\Delta_{11} = \{4, 5, 6, 7\}$. Note that for barges that are with small drafts and can sail in the channels at any time, $\Delta_i = T$.
Γ_k	Set of all pairs of tasks with precedence relationship when sailing in channel k, $\Gamma_k = \{(i, j)\|\text{barge } i \text{ must preceed barge } j \text{ if both sailing in channel } k\}$.
$\mathbf{Z}$	Set of all non-negative integers.
Parameters:	

D	Deadline (time point) of the evacuation.
p_i	Space that barge i occupies when berthed in shelters (in unit sizes). We define the unit size as a 50-meter-long equivalent, hence, barges with lengths within the ranges [0, 50] (m), (50, 75] (m) and (75, 100] (m) occupy spaces of 1, 1.5 and 2 unit sizes in shelters, respectively.
s_i	Minimum safety time (headway) between the start times of barge i and its followers for sailing in the same channel.
N_h	Number of type-h tug boats available for the evacuation.
n_i^1	Number of tug boats (including both HTs and LTs) barge i needs when sailing in the channels of the project site.
n_i^2	Number of tug boats (only HTs) barge i needs when sailing from the outer anchorage of the project site to shelters.
n_{gi}^3	Number of tug boats (including both HTs and LTs) barge i needs when berthing into shelter g.
a_{ki}^1	Time for barge i to sail through channel k in the project site.
a_{gi}^2	Time for barge i to travel from the outer anchorage of the project site to shelter g.
a_{gi}^3	Time for barge i to get berthed into shelter g.
b_k^1	Time for a tug boat to sail back to the site area inside the project site after it finishes tugging a barge out of the site from channel k.
b_g^2	Time for a tug boat to sail back to the outer anchorage of the project site after it finishes tugging a barge to shelter g.
b_g^3	Time for a tug boat to sail back to the outer anchorage of shelter g after it finishes tugging a barge into the berths of the shelter.
C_g^{n}	Largest number of barges that shelter g can accommodate.
C_g^{s}	Capacity of shelter g (in unit sizes).
f_{gi}	Indicator of whether barge i can moor in shelter g which equals 1 if it can and 0, otherwise.
Decision Variables:	
μ_{gi}	1, if barge i is assigned to berth in shelter g and 0, otherwise.
γ_h^1	Number of type-h tug boats assigned to serve in the in-site tugging procedure.
γ_h^2	Number of type-h tug boats assigned to serve in the open-sea tugging procedure.
γ_{gh}^3	Number of type-h tug boats assigned to serve in the in-shelter tugging procedure in shelter g.
ν_{ki}	1, if barge i is tugged out of the site using channel k and 0, otherwise.
x_{it}	1, if barge i starts its in-site tugging at time t and 0, otherwise.
y_{it}	1, if barge i starts its open-sea tugging at time t and 0, otherwise.
z_{it}	1, if barge i starts its in-shelter tugging at time t and 0, otherwise.
α	Start time of the evacuation.
β	Completion time of the evacuation.

3.1.3. The Nonlinear Model

This section proposes the nonlinear MIP model for the BEPP. We first describe the objective and then introduce the constraints in a grouping fashion based on their functions.

Objective. In order to reduce non-working hours of these working barges, the construction party requires the evacuation to start as late as possible. Therefore, the objective of the model is:

$$\max \alpha. \tag{1}$$

Shelter and Tug Boat Assignment Constraints.

$$\sum_{g \in \Theta} \mu_{gi} = 1, \quad \forall i \in \Omega, \tag{2}$$

$$\mu_{gi} \leq f_{gi}, \quad \forall g \in \Theta, \forall i \in \Omega, \tag{3}$$

$$\sum_{i\in\Omega}\mu_{gi}\le C_g^{\mathrm{n}},\quad \forall g\in\Theta, \tag{4}$$

$$\sum_{i\in\Omega}p_i\mu_{gi}\le C_g^{\mathrm{s}},\quad \forall g\in\Theta, \tag{5}$$

$$\gamma_h^1+\gamma_h^2+\sum_{g\in\Theta}\gamma_{gh}^3\le N_h,\quad \forall h\in\Phi. \tag{6}$$

Constraints (2) and (3) ensure that each barge is evacuated to one shelter that is able to harbor it. Besides, Constraints (4) and (5) set limitations for the number of barges that a shelter can harbor and the total space that the shelter has for accommodating barges, respectively. Constraint (6) enforces that the number of tug boats of each type used in the evacuation does not exceed the available number of that type.

Temporal Constraints.

$$\alpha\le\sum_{t\in T}tx_{it},\quad \forall i\in\Omega, \tag{7}$$

$$\beta\ge\sum_{t\in T}tz_{it}+\sum_{g\in\Theta}a_{gi}^3\mu_{gi},\quad \forall i\in\Omega, \tag{8}$$

$$\beta\le D. \tag{9}$$

Constraint (7) enforces the start time of the evacuation should be no later than the time when the first evacuated barge starts its in-site tugging procedure. Constraints (8) indicates that the evacuation ends after all barges have been berthed into shelters. Constraint (9) requires that the evacuation should complete on or before the stipulated deadline.

In-site Tugging Constraints.

$$\sum_{k\in\Psi}\nu_{ki}=1,\quad \forall i\in\Omega, \tag{10}$$

$$\sum_{t\in\Delta_i}x_{it}=1,\quad \forall i\in\Omega, \tag{11}$$

$$\nu_{ki}x_{it}=0,\quad \forall t\in T\setminus\Delta_{ki},\quad \forall k\in\Psi,\forall i\in\Omega, \tag{12}$$

$$\sum_{k\in\Psi}\sum_{i\in\Omega}\sum_{t=\max\{0,t_1-a_{ki}^1-b_k^1+1\}}^{t_1}n_i^1\nu_{ki}x_{it}\le\sum_{h\in\Phi}\gamma_h^1,\quad \forall t_1\in T, \tag{13}$$

$$\sum_{i\in\Omega}\sum_{t=\max\{0,t_1-s_i+1\}}^{t_1}\nu_{ki}x_{it}\le 1,\quad \forall k\in\Psi,\forall t_1\in T, \tag{14}$$

$$\sum_{t\in T}tx_{jt}-\sum_{t\in T}tx_{it}+|T|(2-\nu_{kj}-\nu_{ki})\ge s_i,\quad \forall(i,j)\in\Gamma_k,\forall k\in\Psi. \tag{15}$$

Constraint (10) assigns each barge to exact one channel. Constraints (11) and (12) ensure barges can only be tugged into channels at a time point with suitable tidal conditions. Constraint

(13) ensures that a sufficient number of tug boats are assigned to serve barges in the in-site tugging procedure at each time point. This constraint works as follows. Suppose that barge i is evacuated from channel k (i.e, $\nu_{ki} = 1$). Further, consider that the time to tug barge i in channel k is a^1_{ki} and the time for the tug boats to sail back to the project site after tugging barge i in channel k is b^1_k. Hence, the total time of a tug boat for tugging out barge i through channel k and getting back to the project site is $a^1_{ki} + b^1_k$ which can be taken as the time *occupied* by barge i. Note that exactly n^1_i tug boats are required to serve barge i in the in-site tugging. Therefore, given a time point t_1, if the in-site tugging of barge i starts before or at time t_1 and after $\max\{0, t_1 - (a^1_{ki} + b^1_k) + 1\}$ (the start time cannot be negative), i.e., $\sum_{t=\max\{0,t_1-a^1_{ki}-b^1_k+1\}}^{t_1} x_{it} = 1$, then n^1_i tug boats are still *occupied* by barge i at time t_1. It follows that the number of tug boats *occupied* by barge i at time t_1 is $\sum_{t=\max\{0,t_1-a^1_{ki}-b^1_k+1\}}^{t_1} n^1_i x_{it}$. Now consider that barge i can be evacuated from any channel, the number of tug boats *occupied* by it can generalized to be $\sum_{k\in\Psi} \sum_{t=\max\{0,t_1-a^1_{ki}-b^1_k+1\}}^{t_1} n^1_i \nu_{ki} x_{it}$. Summing up the numbers of tug boats *occupied* at time t_1 by all barges that are evacuated through all channels gives us $\sum_{k\in\Psi} \sum_{i\in\Omega} \sum_{t=\max\{0,t_1-a^1_{ki}-b^1_k+1\}}^{t_1} n^1_i \nu_{ki} x_{it}$, which is the left-hand side of Constraint (13). The right-hand side of this constraint is the number of tug boats assigned to the in-site tugging procedure. Finally, this constraint requires the left-hand side to be no larger than the right-hand side. Constraint (14) defines the minimum headway between two barges that sail consecutively in the same channel. It requires that if barge i is evacuated from channel k and the in-site tugging stars at time t_1 (i.e., $\nu_{ki} x_{it_1} = 1$) then no other barges can be tugged into channel k within the time window $[\max\{0, t_1 - s_i + 1\}, t_1]$ (the start time of an in-site tugging procedure cannot be negative). Thus the minimum headway s_i between barge i and the next barge evacuated from the same channel is ensured. The precedence relationship between two barges for sailing in the same channel is enforced by Constraint (15). To see how the constraint works, consider two cases. First, for each $(i, j) \in \Gamma_k$, $k \in \Psi$ if at least one of ν_{ki} or ν_{kj} equals 0 (i.e., at least one of the barges i and j is not using channel k), then $|T|(2 - \nu_{kj} - \nu_{ki}) \geq |T|$. In this case, Constraint (15) does not remove any feasible solutions to the problem, as the left-hand side of it is always greater than the right-hand side given any feasible x_{jt} and x_{it}. Second, for each $(i, j) \in \Gamma_k$, $k \in \Psi$ if we have $\nu_{ki} = 1$ and $\nu_{kj} = 1$ (i.e., barges i and j are using channel k), then, $|T|(2 - \nu_{kj} - \nu_{ki}) = 0$. In this case, the constraints is equivalent to $\sum_{t\in T} tx_{jt} - \sum_{t\in T} tx_{it} \geq s_i$, which requires that barge i should be evacuated at least s_i unit times prior to barge j (in this way, we ensure that both the precedence relationship and the minimum headway between the two barges are respected).

Open-sea Tugging Constraints.

$$\sum_{t\in T} y_{it} = 1, \quad \forall i \in \Omega, \tag{16}$$

$$\sum_{t\in T} ty_{it} - \sum_{t\in T} tx_{it} \geq \sum_{k\in\Psi} \nu_{ki} a^1_{ki}, \quad \forall i \in \Omega, \tag{17}$$

$$\sum_{g\in\Theta} \sum_{i\in\Omega} \sum_{t=\max\{0,t_1-a^2_{gi}-b^2_g+1\}}^{t_1} n^2_i \mu_{gi} y_{it} \leq \gamma^2_1, \quad \forall t_1 \in T. \tag{18}$$

Constraint (16) ensures that each barge should start its open-sea tugging at one and only one time point. Constraint (17) indicates that the open-sea tugging of a barge should start after the in-site tugging of the barge is completed. In this constraint, $\sum_{t\in T} ty_{it}$ equals the time when barge i starts its open-sea tugging and $\sum_{t\in T} tx_{it}$ equals the time when barge i starts its in-site tugging. On the right-hand side, $\sum_{k\in\Psi} \nu_{ki} a^1_{ki}$ equals the time that barge i should spend in its in-site tugging procedure. Similar to Constraint (13), Constraint (18) ensures that sufficient HTs are assigned for tugging barges in the open sea at any time point.

In-shelter Tugging Constraints.

$$\sum_{t\in T} z_{it} = 1, \ \ \forall i \in \Omega, \tag{19}$$

$$\sum_{t\in T} t z_{it} - \sum_{t\in T} t y_{it} \geq \sum_{g\in\Theta} a^2_{gi}\mu_{gi}, \ \ \forall i \in \Omega, \tag{20}$$

$$\sum_{i\in\Omega} \sum_{t=\max\{0, t_1 - a^3_{gi} - b^3_g + 1\}}^{t_1} n^3_i \mu_{gi} z_{it} \leq \sum_{h\in\Phi} \gamma^3_{gh}, \ \ \forall g \in \Theta, \forall t_1 \in T. \tag{21}$$

Constraint (19) ensures that each barge should start its in-shelter tugging at one and only one time point. Similar to Constraint (17), Constraint (20) enforces the in-shelter tugging of a barge to start after the open-sea tugging is completed for the barge. Similar to Constraints (13) and (18), Constraint (21) ensures that there are sufficient tug boats for berthing barges in each shelter at each time point.

Variable Domains.

$$\alpha, \beta \geq 0, \tag{22}$$

$$\gamma^1_h, \gamma^2_h \in \mathbf{Z}, \ \ \forall h \in \Phi, \tag{23}$$

$$\gamma^3_{gh} \in \mathbf{Z}, \ \ \forall g \in \Theta, \forall h \in \Phi, \tag{24}$$

$$\mu_{gi} \in \{0, 1\}, \ \ \forall g \in \Theta, \forall i \in \Omega, \tag{25}$$

$$\nu_{ki} \in \{0, 1\}, \ \ \forall k \in \Psi, \forall i \in \Omega, \tag{26}$$

$$x_{it}, y_{it}, z_{it} \in \{0, 1\}, \ \ \forall i \in \Omega, \forall t \in T. \tag{27}$$

The first constraint requires the temporal variables α and β to be non-negative. Constraints (23) and (24) ensure γ's are non-negative integers. The last three constraints define binary variables.

3.1.4. Model Linearization

Model M1 is nonlinear due to the multiplications among decision variables in Constraints (12), (13), (14), (18), and (21). However, most off-shelf solvers are either unable to solve or can only obtain less satisfactory solutions for nonlinear models. Hence, we linearize M1 by using the method proposed in Appendix A.

3.2. Complexity of the Problem

This section demonstrates that the BEPP is NP-hard in the strong sense. To do this, we show that the decision version of the BEPP is strongly NP-hard. That is, given the parameters regarding the barges, evacuation recourses, and evacuation procedures, it cannot be determined in polynomial time or even in pseudo-polynomial time whether the objective value α is no less than a given constant $\overline{\lambda}$ unless P=NP. We prove that the decision version of the BEPP is NP-hard in the strong sense by reducing the Three-Machine Flow Shop Scheduling Problem (3M-FSP) into a decision version of the BEPP. The 3M-FSP has been proved to be strongly NP-hard by Garey et al. (1976).

The theoretical complexity of the BEPP is proposed in Theorem 1.

Theorem 1. *The BEPP is NP-hard in the strong sense.*

We prove the theorem by transforming the decision version of the 3M-FSP to the decision version of the BEPP, and the detailed proof is given in Appendix B.

Remark 1. *The BEPP remains NP-hard in the strong sense, even when each barge requires only one tug boat in each procedure and each procedure has only one tug boat.*

In the proof for Theorem 1, we reduce the 3M-FSP to the BEPP. In the 3M-FSP, each job is processed on only one machine in three stages and there is only one machine at each stage. Hence, the result follows directly from Theorem 1.

It is mentionable that although we can reduce the 3M-FSP to the BEPP, algorithms for the 3M-FSP or other Flow Shop Scheduling Problems (e.g., Chen et al., 1996; Ben-Daya and Al-Fawzan, 1998) can hardly be applied to solve the BEPP. This is because, in Flow Shop Scheduling Problems, the number of machines is fixed in each processing stage while in the BEPP the number of tug boats in each stage is a decision variable. In addition, in the Flow Shop Scheduling Problems, each task requires only one machine to process in each stage, while in the BEPP, synchronization of tug boats is required such that several tug boats are required to serve a barge at the same time. Note that even a single-stage Parallel Machine Scheduling Problem with such synchronization requirements is very hard to solve (refer to Du and Leung, 1989; Wu and Wang, 2018) and that the BEPP involves multiple stages.

3.3. The Solution Procedure

We propose a heuristic algorithm to solve the BEPP because of the high complexity of the problem (see Section 3.2). In addition, the special structure and features of the problem (as illustrated in Section 3.3.1) also enable us to propose efficient heuristic strategies for identifying a high-quality solution. The framework of the algorithm is given in Section 3.3.2. The key comments in the algorithm are explained in Section 3.3.3.

3.3.1. Observations on the Problem

We find that several features of the BEPP faced by HKIA can be utilized for developing an efficient heuristic algorithm. These features are obtained from our discussions with HKIA, our analysis of the problem, and the results of earlier numerical experiments. We present these features as follows:

Observation 1. *The distance between a shelter and the project site is the most decisive factor, among others, for selecting shelters to accommodate barges. Shelters with shorter distances have higher priorities.*

Shorter distance means less workload for the tug boats serving in the open-sea tugging procedure. Observation 1 is also based on the fact that the numbers of required tug boats in the in-shelter tugging procedure for each barge are similar in different shelters (of Type I) and so are the berthing times for each barge in these shelters, and that these berthing times are usually shorter than the times needed in the open-sea tugging procedure.

Observation 2. *The sequences (of starting the in-site tugging procedure) among barges evacuated to different shelters have greater impacts on the total evacuation time than the sequences (of starting the in-site tugging procedure) among barges assigned to the same shelter.*

For safety consideration, barges have to be tugged at a very low and constant speed when sailing through the channels of the project site and when getting berthed into a shelter. Besides, during the open-sea tugging, HTs also should maintain a constant speed. Therefore, for barges with the same assigned shelters, they spend almost identical times in in-site tugging, open-sea tugging, and in-shelter tugging. If barges assigned to the same shelter are also of similar draft and size (in many cases they are), then the evacuation processes of them are identical, and the sequences among them barely have any impact on the total evacuation time. However, due to the different distances between shelters and the project site, the times for tugging barges assigned to different shelters across the open sea vary from one to another. Hence, the sequences among barges assigned to different shelters may significantly affect the utilization of tug boats used in the open-sea tugging, which further affects the total evacuation time.

Observation 3. *In the BEPP faced by HKIA, the three channels $k \in \Psi$ are identical such that they share the same a^1_{ki}, $i \in \Omega$, b^1_k and Δ_k. Now suppose the following decisions are given: (i) the assignment of barges among shelters, (ii) the assignment of tug boats among evacuation procedures and shelters, and (iii) the sequences to start evacuating the barges. Then, the List Scheduling rule (refer to Schutten, 1996) generates (i) the optimal scheduling of channels for evacuating barges in the in-site tugging procedure, and (ii) the optimal scheduling of tug boats for tugging barges in each procedure.*

The List Scheduling rule is a term that stems from machine scheduling problems. It assigns machines with the earliest ready times to tasks in a given sequence. We adopt the same idea when assigning channels and tug boats to barges, by taking channels and tug boats as machines and the through-channel sailing and towage services of barges as tasks in various evacuation procedures. The optimality of the List Scheduling rule for scheduling channels and tug boats with given service sequences is easy to verify since any other method leads to unnecessary delays for the barges.

To see how the List Scheduling rule works, consider the following example. Suppose at a time point $t = 10$, there are three barges (i_1, i_2, and i_3) waiting to be evacuated to a set of shelters in the outer anchorage of the project site. The sequence for evacuating these barges is known and is given by $i_2 \to i_1 \to i_3$. The numbers of tug boats required by these barges in the open-sea tugging are $n^2_{i_1} = 1$, $n^2_{i_2} = 2$, and $n^2_{i_3} = 1$, respectively. Suppose that these three barges are evacuated to the same shelter, and the times for the open-sea tugging of them are all two unit times. Three tug boats (denoted by 1, 2, and 3) are serving in the open-sea tugging, and the available times of them (i.e., times ready to tug a barge) are 10, 11, and 12 for tug boats 1, 2, and 3, respectively. The time to return to the outer anchorage of the project site from the shelter of these barges is one unit time. We now apply the List Scheduling rule to schedule the tug boats to serve the barges, and the resultant schedule for the tug boats is given in Figure 6. In this schedule, we always assign tug boats with the earliest ready times to the first barge in a given sequence that is still waiting to be tugged.

3.3.2. Algorithm Framework

The algorithm, as shown in Figure 7, solves the problem in a two-stage fashion. In stage one, we first assign each barge to a shelter and then obtain an estimation of the makespan of the evacuation by solving a truncated version of the BEPP (denoted by T-BEPP). In the T-BEPP, the shelter assignment result is given and barge draft and tidal conditions are not considered (i.e., each barge can be tugged into the channels at any time). We then take tidal conditions into consideration, and barges can only sail in the channels within certain time windows. In the second stage, we seek to find a feasible makespan in a trial-and-error manner. To be specific, in each iteration we verify the feasibility of the incumbent makespan by solving the problem with the derived evacuation start time (which equals the evacuation deadline minus the incumbent makespan). If

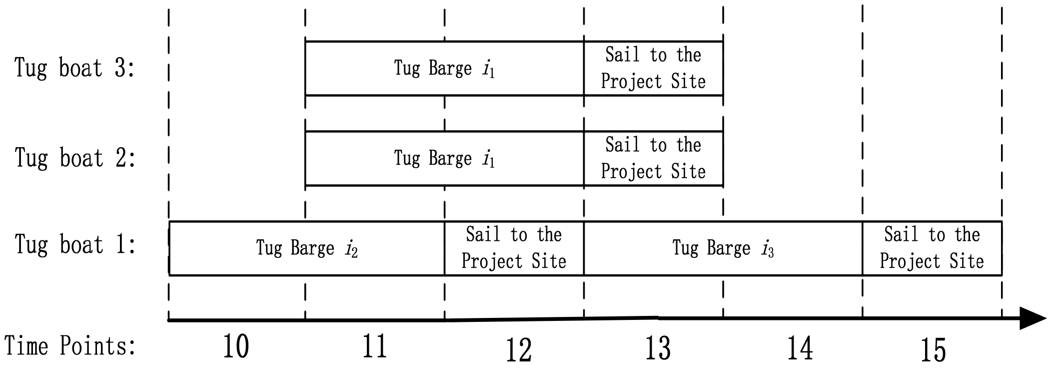

Figure 6: The Gantt Chart for the schedule of the tug boats.

the incumbent makespan is feasible (i.e., the resultant evacuation completion time is no later than the deadline), the algorithm stops; otherwise, the incumbent makespan is increased by a certain length of additional time (denoted by AT), and we start a new iteration with the updated makespan.

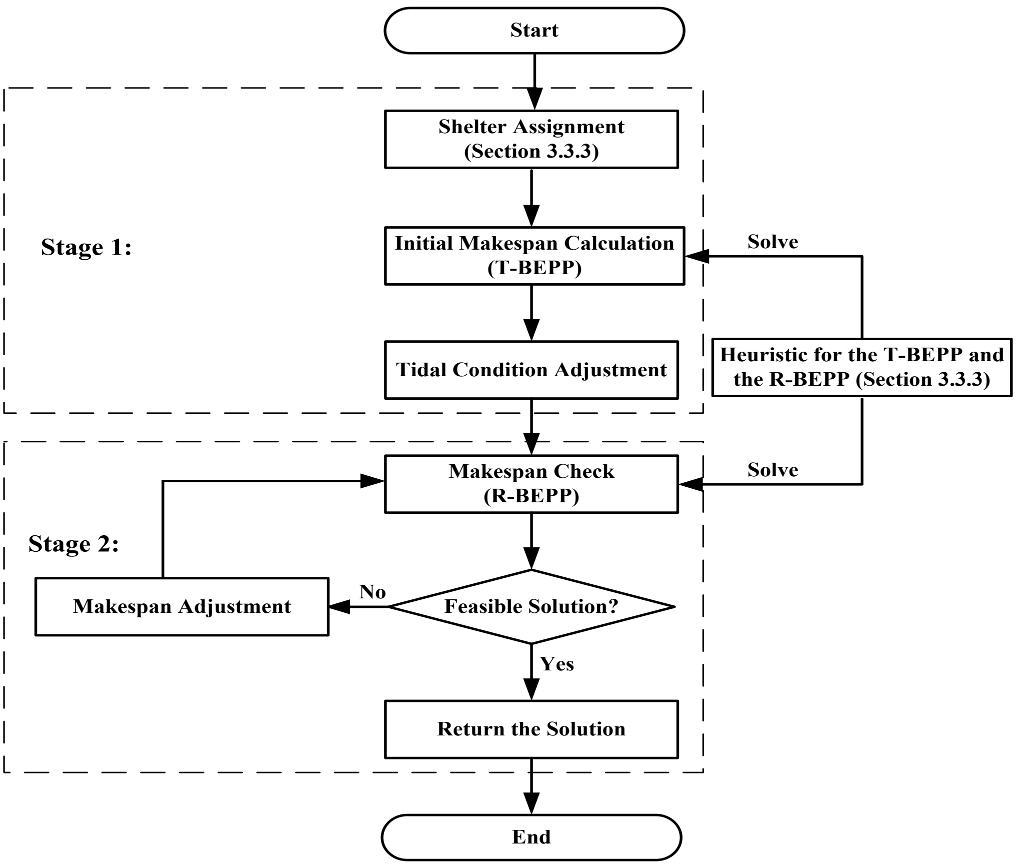

Figure 7: The framework of the solution algorithm.

The efficiency of the solution framework largely depends on the performance of the algorithm embedded in the framework for estimating (in Stage 1) and verifying (in Stage 2) the makespan. The algorithm, by nature, solves the T-BEPP in Stage 1 and another revised BEPP (denoted by R-BEPP) in Stage 2. In the R-BEPP, the start time of the evacuation is given and the deadline constraint is removed, and the objective is to minimize the makespan of the evacuation. Note that the R-BEPP is also a truncated version of the BEPP, since the shelter assignment plan (obtained in Stage 1) is fixed. Also, note that the T-BEPP and the R-BEPP differ in the settings of evacuation start time and suitable time windows for tugging barges into the channels. In particular, in the T-BEPP the evacuation starts at time point 0 and barges can be tugged into the channels at any time (i.e., all barges have the same time window spanning through time 0 to D, since the constraints regarding tidal conditions and barge drafts are not considered). By contrast, in the R-BEPP the evacuation starts at a particular starting time (which is derived from the estimated makespan in Stage 1 or given by the Makespan Adjustment procedure in Stage 2) and barges can only be tugged into the channels within suitable time windows (i.e., tidal conditions and barge drafts are considered). To solve the T-BEPP and the R-BEPP, we develop a heuristic which takes advantage of the special structure and features of the problem. In the following section, we explain the method for assigning the shelters to barges and then introduce the heuristic for solving the T-BEPP and the R-BEPP.

3.3.3. Components in the Solution Algorithm

In this section, we introduce the shelter assignment method and the heuristic for solving the T-BEPP and the R-BEPP. The rich details of them are given in Appendix C.

Shelter Assignment. The first step to solve the BEPP is shelter assignment. For shelter assignment, we apply the procedure as shown in Algorithm 1 in Appendix C.1. The procedure, which takes advantage of Observation 1, assigns barges to the closest shelter (i.e., shelters that can be reached in the shortest time) with spare capacity. In addition, in the assignment, priorities are given to barges with larger sizes (larger barges may require more tug boats in the open-sea tugging). We design such a procedure in order to reduce the workload of tug boats in the open-sea tugging.

A Heuristic for Solving the T-BEPP and the R-BEPP. We develop a two-stage local-search-based algorithm to solve the T-BEPP and the R-BEPP. As shown in Figure 8, the heuristic is a combination of two meta-heuristics (i.e., a Tabu Search algorithm [TS] and a Simulated Annealing algorithm [SA]). At the primary level, the TS is used to identify the optimal tug boat assignment pattern among evacuation procedures and shelters for the T-BEPP or the R-BEPP. Simultaneously, the SA, which is embedded into the TS at the secondary level, generates the best evacuation sequence among the barges under a given assignment pattern of the tug boats. Meanwhile, given the tug boat assignment pattern and the evacuation sequence, the makespan of the T-BEPP or the R-BEPP is obtained by the resource scheduling procedure. In Appendix C.2, we provide the details regarding the TS, the SA, as well as the resource scheduling procedure. We structured the algorithm in this way because we found that the assignment of tug boats among different evacuation procedures and shelters has the most significant impact upon the makespan of both the T-BEPP and the R-BEPP, and that the evacuation sequence of the barges also affects the makespan, though the impact is relatively weaker.

4. Numerical Experiments

In this section, we perform extensive computational experiments to verify the applicability and effectiveness of the proposed model and solution method. In addition, we provide a case study in which the proposed algorithm is used to solve the real BEPP faced by HKIA. The experiments

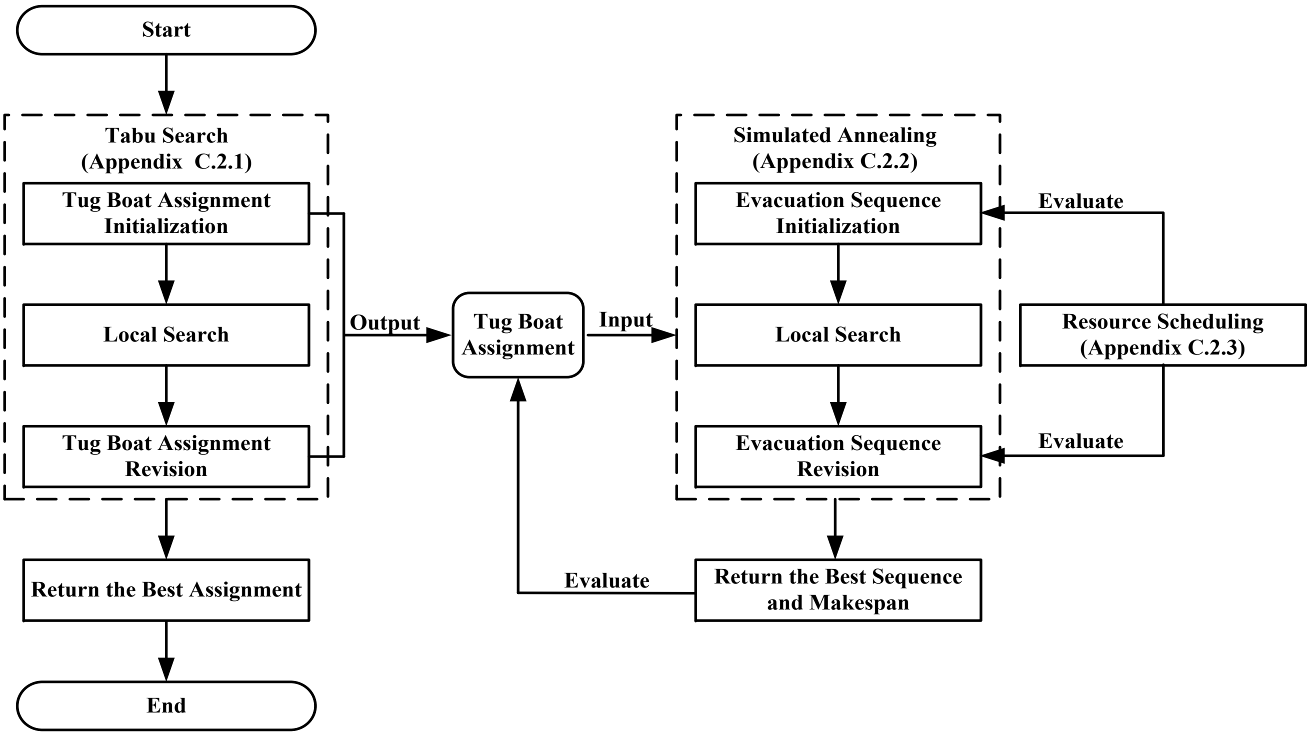

Figure 8: An overview of the heuristic for solving the T-BEPP and the R-BEPP.

are performed on two sets of instances (Set A and B) which have different input parameters. The instances in Set A are small scale instances with 5 to 20 barges, while those in Set B are instances with large (practical) scales with 50 to 100 barges. Instances in Set A are solved by the heuristic algorithm proposed in the previous section (for notational simplicity, we denote the method by HA) and CPLEX using the linearized MIP model. For instances in Set B, we solve them by the HA and three other heuristics which will be described in Section 4.2. All the experiments are coded in C++ and are conducted on an Intel Core i7 2.50 GHz PC with 8 GB RAM. CPLEX 12.6 is used as the MIP solver for instances in Set A.

4.1. Instance Generation

In order to test the performances of the proposed model and algorithm, we generated 20 instances in Set A and 30 instances in Set B based on real-world cases. The input data involving the barges to be evacuated and the evacuation-supporting resources (i.e., tug boats, channels, and shelters) in these instances are generated as follows.

To begin with, for instances in Set A and Set B, the numbers of barges to be evacuated (i.e., $|\Omega|$) are chosen from $\{5, 10, 15, 20\}$ and $\{50, 60, 70, 80, 90, 100\}$, respectively. For each input of $|\Omega|$, we generate 5 random instances, leading to 50 instances in total. For notational simplicity, we denote an instance by $|\Omega|$-I, where $|\Omega|$ is the number of barges to be evacuated in the instance and $I \in \{1, 2, 3, 4, 5\}$ represents the index for the instance in a group of instances that share the same $|\Omega|$.

As for the sizes of these barges, we set the numbers of barges with small sizes (less than 50m), medium sizes (50m–75m) and large sizes (75m–100m) in an instance to be $\lfloor 0.5|\Omega| \rfloor$, $\lfloor 0.4|\Omega| \rfloor$, and

$|\Omega| - \lfloor 0.5|\Omega| \rfloor - \lfloor 0.4|\Omega| \rfloor$, respectively. Barges with different sizes may occupy different sizes of spaces in the shelters and also have different requirements regarding the numbers of tug boats used in various evacuation procedures. Table 2 shows these parameters in detail. In addition, the drafts of the barges are generated in a way such that around 10% barges are of large drafts and the remaining 90% are of small drafts. For large draft barges, their drafts are randomly generated using the distribution $U(6, 9]$ and drafts for those with small drafts are randomly generated using the distribution $U[2, 6]$.

Table 2: Parameters of the barges in the instances.

Size	Shelter space[1,2]	Number of required tug boats		
		In-site tugging[3]	Open-sea tugging[4]	In-shelter tugging[3,5]
Small	1.0	1	1	1
Medium	1.5	1	2	1
Large	2.0	2	3	1

Note[1]: The number of unit sizes a barge occupies when berthing in a shelter.
Note[2]: One unit size is a 50-metre-long equivalent.
Note[3]: Both HTs and LTs can be used in in-site tugging and in-shelter tugging.
Note[4]: Only HTs can be used in open-sea tugging.
Note[5]: Only for Type I shelters.

The parameters regarding the evacuation-supporting resources are set as follows. First, the number of tug boat is fixed at 10 for all small instances in Set A, and there are 6 HTs and 4 LTs. For instances in Set B, we set the number of tug boats to be 50, including 30 HTs and 20 LTs. Second, we set the number of channels $|\Psi| = 1$ and $|\Psi| = 3$ for instances in Set A and B, respectively. For the precedences among barges for sailing into these channels, we randomly select $\lceil 0.1|\Omega| \rceil$ barges each of which blocks other barges from entering a channel that is also randomly selected. Third, the parameters of shelters are generated as follows. There are two types of shelters (Type I and Type II). For each instance in Set A, the number of Type I shelters is set to be 2, and for those in Set B, the number of Type I shelters is set to be 5. For each Type I shelter, the capacity (C^{s}_g) of it is randomly and uniformly generated within the range $[\frac{\bar{S}}{M}, \frac{2\bar{S}}{M}]$, where $\bar{S}$ is the total space required by all barges and M is the number of Type I shelters. Besides, no limits are set for the total number of barges a Type I shelter can harbor, and all shelters can harbor barges with all sizes. In addition, there is one Type II shelter which can harbor at most 1 and 3 barges (i.e., $C^{\mathrm{n}}_g = 1$ and $C^{\mathrm{n}}_g = 3$) for each instance in Set A and B, respectively. Capacities of the Type II shelters are set to be $C^{\mathrm{s}}_g = 2C^{\mathrm{n}}_g$.

The temporal parameters involved in the instances are generated as follows. First, in all instances, we set the unit time to be 15 minutes, that is, a day contains 96 unit times. The deadline is set as a time point in a day that is randomly selected from the range $[1, 96]$. The evacuation can start as early as 60 hours before the deadline, that is, $|T| = 240$. Then, we simulate the tidal pattern in all channels in a day using the sine curve $9 + 3\sin\frac{\pi h}{24}$, where $h = \{1, ..., 96\}$ represents the length (unit times) after zero o'clock in a day. Besides, the minimum safety time (headway) between the start times of a barge and its followers for sailing in the channels (s_i) is set to be 15 minutes for all barges in all instances. Finally, the time for evacuating barges in each evacuation procedure (denoted by "Barge time" in Table 3) and the time for tug boats to get ready for the next task after completing a task (denoted by "Tug boat time" in Table 3) are randomly generated according to the ranges shown in Table 3. Note that in each instance, we assume that there is only one route between the project site and each shelter, and all barges in the instance can sail in the route (i.e., height limitations of routes are not considered). In addition, all barges in one instance share the same Barge times in the in-site tugging and the in-shelter tugging procedures, and they also spend the identical time if tugged to the same shelter in the open-sea tugging procedure. However, the times for the open-sea tugging procedure of a barge can vary among different destination shelters. The tug boat times in an instance are also set in a similar way, and for the open-sea tugging procedure, the time for a tug boat to return to the project site from a shelter is set to be around $\frac{1}{4}$ of the barge time in the open-sea tugging procedure bound for this shelter.

Table 3: Ranges of the times of evacuation procedures in the instances.

Item	In-site tugging	Open-sea tugging		In-shelter tugging[3]
		Type I shelters	Type II shelters	
Barge time	[6,10][1,2]	[8,40]	[8,12]	[8,12]
Tug boat time	[1,1][2]	[2,10]	[2,3]	[1,2]

Note[1]: All ranges are shown in unit times.

Note[2]: We set identical Barge times and identical Tug boat times for all channels.

Note[3]: Only for Type I shelters.

4.2. Algorithm Settings

To compare the performance of the HA, we devised another three heuristics, each of which is a simplified version of the HA. In the first heuristic, which is denoted by SHA1, we remove the SA for sequencing barges from the HA. Instead, given a tug boat assignment pattern, the sequence of barges in the SHA1 is generated using the initial sequencing procedure as shown in Algorithm 3. Meanwhile, in the second heuristic, we remove the TS for assigning tug boats from the HA, and the tug boats are assigned among different procedures using the procedures that are used to initialize the assignment patterns in the HA (see Section Appendix C.2.1). We denote the second heuristic by SHA2. Finally, in the last heuristic, we replace both the TS for tug assignment and the SA for barge sequencing by two simple local search algorithms. In the two local search algorithms, we use the same neighborhood construction strategies as in the TS and SA. In each iteration of a local search algorithm, the algorithm searches within a corresponding neighborhood and the incumbent solution is updated only when a better one (i.e., the one with a smaller makespan) is found. We denote the last heuristic by SHA3.

The parameters used in the algorithms are set as follows. To begin with, in all of the four heuristics, the "search length" AT that will be added to an infeasible incumbent makespan is set to be 15 mins (one unit time). Then, we set the parameters used in the TS that is incorporated into the HA and the SHA1 and the SA that is used in the HA and SHA2 as follows. First, in the TS, we set the "penalty multiplier" PM for repeating tuples to be 0.6 and the "tabu length" TL to be 5. Also, $\kappa_{\max}$, which controls the number of iterations in one run of the TS, is set to be 30. Second, in the SA, the "starting temperature" τ_0, the "temperature reduction rate" ρ, and the Boltzmann constant bc are set to be 50, 0.95, and 0.11, respectively, and the algorithm stops after 100 iterations. Finally, we run each heuristic algorithm 30 times to solve each instance and set the time limit of CPLEX for solving the instances in Set A to be 3600 seconds.

4.3. Computational Results

We report the results of the numerical experiments in the section. In particular, the results of instances in Set A are reported in Section 4.3.1 and the results of instances in Set B are given in Section 4.3.2.

4.3.1. Results of Instances in Set A

Instances in this part were solved by the HA and CPLEX. The results obtained by the two solution methods for these instances are reported in Table 4. In the table, the performance of the HA for solving each instance is measured by (i) the average objective value (α) in the 30 runs, which is demonstrated in Column "AS" (ii) the objective value (α) of the best solution in the 30 runs, which is demonstrated in Column "BS", (iii) the objective value (α) of the worst solution in the 30 runs, which is demonstrated in Column "WS", and (iv) the average computational time in the 30 runs, which is demonstrated in Column "Time". As for CPLEX, we report the objective value of the best solution found by CPLEX (in Column "Solution") and the time it took to solve the instance (in Column "Time"). Note that the solution delivered by CPLEX for an instance may not be optimal if the computational time is 3600.00 seconds (the time limit). Besides, "–" in the table denotes that CPLEX failed to obtain a feasible solution for an instance within the time limit. In addition, for each instance, the difference between the objective value of the solution obtained by CPLEX and the average objective value obtained by the HA is reported in the last column.

Table 4: Results of instances in Set A.

Instance	HA				CPLEX		
	AS	BS	WS	Time(s)	Solution	Time(s)	GAP
5-1	198.00	198	198	0.11	202	9.99	4.00
5-2	168.00	168	168	0.08	171	43.50	3.00
5-3	189.00	189	189	0.03	189	23.31	0.00
5-4	185.00	185	185	0.02	188	86.08	3.00
5-5	180.00	180	180	0.03	187	24.53	7.00
10-1	109.00	109	109	0.67	109	3600.00	0.00
10-2	99.00	99	99	0.02	98	3600.00	-1.00
10-3	134.00	134	134	0.03	136	3600.00	2.00
10-4	155.00	155	155	0.03	159	3600.00	4.00
10-5	154.00	154	154	0.02	151	3600.00	-3.00
15-1	104.00	104	104	0.03	–	3600.00	–
15-2	106.00	106	106	0.05	–	3600.00	–
15-3	128.00	128	128	0.03	84	3600.00	-44.00
15-4	140.00	140	140	0.06	95	3600.00	-45.00
15-5	129.00	129	129	0.18	92	3600.00	-37.00
20-1	46.00	46	46	0.04	–	3600.00	–
20-2	82.10	83	82	0.74	–	3600.00	–
20-3	37.00	37	37	1.53	–	3600.00	–
20-4	61.00	61	61	0.02	–	3600.00	–
20-5	29.00	29	29	0.49	–	3600.00	–

As shown in the table, CPLEX can only solve the smallest instances (i.e., instances with 5 barges) to optimum within 3600 seconds. For these instances, the HA is able to obtain optimal or near-optimal solutions. For the instances with 10 barges, the solutions provided by the two methods are also very similar. However, when the number of barges in an instance reaches 15, CPLEX fails to find a feasible solution or can only provide a solution that is significantly worse than the solution provided by the HA. For instances with 20 barges, CPLEX cannot deliver feasible solutions within the time limit. As for the solution time, the HA solves almost all instances within 1 second, while CPLEX reaches the time limit (3600s) in most of the instances. Therefore, the HA outperforms CPLEX in terms of the solution speed for solving all the instances in this set, and it outperforms CPLEX in terms of both speed and solution quality for instances with 15 and 20 barges.

4.3.2. Results of Instances in Set B

This section reports and compares the performances of four heuristics (i.e., the HA, the SHA1, the SHA2 and the SHA3) for solving the large-scale instances in Set B. In particular, in Table 5, we report the average objective value (α), the objective value (α) of the best solution and the objective value (α) of the worst solution for each instance obtained in 30 runs of the four methods. We then derive the improvements (in %) of the solutions obtained by the HA against the solutions obtained by other heuristics which is calculated by $100\frac{\alpha_{HA}-\alpha_{SA}}{\alpha_{SA}}$. In this equation, α_{HA} (or α_{SA}) is the average objective value of the solutions, the objective value of the best solution, or the objective value of the worst solution obtained by the HA (or another heuristic: SHA1, SHA2, or SHA3) for an instance. The averages of such improvements for instances in each group are reported in Table 6. Note that the averages in this table are calculated after removing $100\frac{\alpha_{HA}-\alpha_{SA}}{\alpha_{SA}}$ with $\alpha_{SA} \leq 0$.

In practice, a later starting time for an evacuation means less loss in the productivity of barges and lower renting costs for the construction party. Hence, to further compare the performances of the algorithms, we calculate the estimated savings in the renting cost obtained by the HA against other heuristics. Particularly, given the α_{HA} and α_{SA} as defined above for an instance, we estimate the savings by $|\Omega|\overline{rc}(\alpha_{HA} - \alpha_{SA})$. Here $|\Omega|$ equals the number of barges in the instance and $\overline{rc}$ is the estimated unit time renting cost (in US dollars) per barge. According to HKIA's case we set $\overline{rc} = 30$. The averages of such savings for instances in each group are reported in Table 7. Finally, the average computational times (in one run) of each method for each group of instances are reported in Table 8 (instances with the same number of barges are assembled in one group).

As shown in Table 5, among the 30 instances, the HA outperforms the other 3 methods for 29 instances in terms of the average solution. It also manages to find the best solutions for all of the instances. In addition, the HA also reports the best worst solutions for 28 of the 30 instances. Therefore, in terms of the solution quality, the HA outperforms all the other solution methods for solving instances in Set B.

We can see from Table 6 that the HA reports improvements (on average) against all the other algorithms in terms of "Average Solutions", "Best Solutions", and "Worst Solutions" for instances with all sizes. Therefore, our algorithm is consistently better than the other algorithms. Since the SHA1, SHA2, and SHA3 can be viewed as the simplified versions of the HA, the results also demonstrate that the two-stage structure and the algorithm used in each stage are valid for improving the performance of the HA. It is also mentionable that the improvements generally grow with the sizes of the instances. This indicates that our algorithm generates greater benefits against other algorithms in applications with large scales.

As shown in Table 7, compared with other algorithms, the HA is capable of generating considerable savings in an evacuation for a construction party. When compared with the second-best solution algorithm (i.e., the SHA3), the HA averagely brings at least $4,500$ dollars' reduction in the renting cost in terms of the average solutions in all instance groups. As expected, the savings increase with the scale of the instances. In addition, the superiority of the HA is more obvious when it comes to the worst solutions. This indicates that the HA secures relatively even better solutions in the worst case.

When it comes to the solution speed, as shown in Table 6, the two two-stage heuristic methods (i.e., the HA and the SHA3) are significantly slower than the one-stage methods (i.e., the SHA1 and the SHA2). Nevertheless, the average times for the HA to solve instances in different groups are all less than 510 seconds (less than 10 minutes). Therefore, the HA is able to instantly provide feasible solutions for real applications. It is also mentionable that for most of the instances, the HA takes less time to converge than the SHA3 does.

Table 5: Results of instances in Set B.

Instance	Average Solution				Best Solution				Worst Solution			
	HA	SHA1	SHA2	SHA3	HA	SHA1	SHA2	SHA3	HA	SHA1	SHA2	SHA3
50-1	169.66*	166.00	132.00	167.00	171*	166	132	167	168*	166	132	167
50-2	140.28*	136.00	111.00	136.55	141*	136	111	137	139*	136	111	136
50-3	114.41*	104.00	82.38	107.41	115*	104	83	109	112*	104	73	105
50-4	157.00*	149.00	117.76	149.41	157*	149	120	150	157*	149	111	149
50-5	152.00*	152.00*	126.93	152.00*	152*	152*	127	152*	152*	152*	126	152*
60-1	121.00*	110.00	69.86	117.86	121*	110	72	120	121*	110	69	115
60-2	132.10*	129.00	103.07	131.00	133*	129	105	131	132*	129	102	131
60-3	154.38*	148.97	123.03	153.00	156*	149	124	153	154*	148	123	153
60-4	136.21*	128.00	90.24	128.07	138*	128	92	129	136*	128	89	127
60-5	141.93*	139.00	87.00	139.69	142*	139	87	140	141*	139	87	139
70-1	114.59*	114.00	80.00	114.00	120*	114	80	114	114*	114*	80	114*
70-2	135.03*	130.10	111.14	132.34	136*	133	112	133	135*	130	104	114
70-3	107.00*	106.00	50.21	107.00*	107*	106	54	107*	107*	106	42	107*
70-4	113.41*	102.00	85.00	106.48	114*	102	85	114*	112*	102	85	103
70-5	133.86*	130.00	115.00	133.34	134*	130	115	134*	133*	130	115	133*
80-1	107.93*	103.00	78.17	107.00	108*	103	84	107	106	103	76	107*
80-2	128.10*	117.00	85.97	118.76	129*	117	86	125	128*	117	85	118
80-3	97.24	97.00	85.24	98.00*	98*	97	86	98*	96	97	84	98*
80-4	104.90*	100.97	44.38	99.03	105*	101	46	103	104*	100	43	94
80-5	101.69*	100.03	91.17	101.10	102*	101	93	102*	101*	100	76	100
90-1	73.55*	53.00	47.48	66.03	74*	53	48	71	73*	53	46	54
90-2	50.45*	40.83	20.59	46.55	51*	41	31	49	47*	36	16	43
90-3	59.79*	42.00	12.79	49.45	61*	42	17	59	46*	42	-5	42
90-4	98.00*	92.00	90.00	92.83	98*	92	90	93	98*	92	90	92
90-5	98.28*	95.00	71.62	96.00	100*	95	72	96	98*	95	70	96
100-1	90.00*	86.00	17.00	89.79	90*	86	17	90*	90*	86	17	89
100-2	106.00*	103.00	50.14	95.00	106*	103	59	95	106*	103	47	95
100-3	93.00*	90.00	54.00	91.00	93*	90	54	91	93*	90	54	91
100-4	24.00*	17.83	-7.76	21.24	26*	18	5	22	20*	13	-10	15
100-5	81.83*	77.00	56.17	80.66	82*	77	57	81	81*	77	55	79
NBR	29	1	0	3	30	1	0	7	28	2	0	6

Note. We use "*" to mark the best result obtained by the four solution methods.
Note. The last row reports the total number of best results (i.e., entries marked with "*") in each column.

Table 6: Improvements (in %) of the HA against other heuristics.

Instance	Average Solution			Best Solution			Worst Solution		
Size ($\|\Omega\|$)	SHA1	SHA2	SHA3	SHA1	SHA2	SHA3	SHA1	SHA2	SHA3
50	4.15	29.37	3.18	4.53	29.13	3.10	3.29	33.60	2.97
60	4.91	48.18	2.47	5.55	46.75	2.55	4.81	48.97	3.03
70	3.88	45.54	1.89	4.66	44.04	1.50	3.38	54.90	5.43
80	4.01	49.81	2.89	4.22	46.09	1.22	3.26	55.82	3.43
90	22.94	122.69	9.72	24.21	85.06	4.25	17.50	75.33	12.52
100	10.36	164.68	5.69	12.37	209.03	6.64	13.99	168.61	10.15

Table 7: Savings (in thousand US dollars) obtained by the HA against other heuristics.

Instance Size ($\|\Omega\|$)	Average Solution			Best Solution			Worst Solution		
	SHA1	SHA2	SHA3	SHA1	SHA2	SHA3	SHA1	SHA2	SHA3
50	7.90	48.98	6.29	8.70	48.90	6.30	6.30	52.50	5.70
60	11.04	76.47	5.76	12.60	75.60	6.12	10.80	77.04	6.84
70	9.15	68.27	4.50	10.92	69.30	3.78	7.98	73.50	12.60
80	10.49	74.37	7.66	11.04	70.56	3.36	8.64	82.08	8.64
90	30.91	74.30	15.77	32.94	68.04	8.64	23.76	63.45	18.90
100	12.60	145.14	10.28	13.80	123.00	10.80	12.60	147.75	12.60

Table 8: Average computational time (in seconds) for instances in Set B.

Instance Size ($\|\Omega\|$)	HA	SHA1	SHA2	SHA3
50	42.55	0.12	0.31	47.42
60	100.64	0.25	0.34	230.21
70	147.40	0.26	0.60	98.51
80	321.20	0.42	1.61	440.02
90	290.00	0.35	1.32	309.27
100	502.92	0.49	0.86	747.46

4.4. Case Study

To construct the 3RS, HKIA is now conducting a land reclamation project. The project, which will add approximately 650 hectares of new land for HKIA in only 4 years, is one of the largest land reclamation projects in the world. There are 102 barges working for the land reclamation project of HKIA. The detailed input of the case is presented in Appendix D.1. In June 2018, HKIA performed an evacuation of the 102 barges to protect them from an approaching storm. The details are as follows. On 9.00 a.m. on the first day, HKIA forecast a typhoon that threatened the safety of barges in the 3RS project site was expected to arrive at the project site in 60 hours. Hence, all barges must be evacuated to shelters before 9.00 p.m. on the third day. Knowing this, the managers in charge of the evacuation then generated a whole evacuation plan based on the rules of thumb. It took the managers more than 10 hours to work out the complete evacuation plan. The evacuation started at 9.00 a.m. on the second day, and the 102 barges were evacuated in 36 hours from the 3RS project site to shelters.

We solve the same BEPP faced by HKIA by the HA (by using the data provided by HKIA). The algorithm was run 30 times using the settings as presented in Section 4.2. It converged to the same optimal start time (2.30 p.m. on the second day or $\alpha = 118$) and the same optimal makespan (30.5 hours) in all the 30 runs. The computational times in the 30 runs are all less than 770 seconds. Detailed shelter and tug boat assignment results obtained by the algorithm are shown in Appendix D.2. In comparison, our proposed algorithm solves the problem in a much shorter time (less than 13 minutes) and the derived evacuation time is also considerably shorter. Hence, by using the algorithm, we not only greatly lessen the burden on managers in charge of the evacuation, but also save a lot of cost by enabling the evacuation to start at a later time.

4.4.1. Impacts of Traffic Control in Shelters

In this section, we consider the scenario where the traffic flows in the shelters (of type I) are strictly controlled to avoid congestion. To model such control, we require that (1) barges have to queue up when entering a shelter and (2) there should be a minimum headway between any two barges that sail into the same shelter consecutively. We denote such a minimum headway by MH, which indicates that for two barges that sail into a shelter consecutively, the following barge can starting sailing into the shelter no earlier than MH unit times after the leading barge starts sailing into the shelter.

To evaluate the impact of traffic control in shelters on the performance of the HA, we consider three settings of MH in the BEPP faced by HKIA, including $MH = 0$, $MH = 1$ and $MH = 2$. Note that when $MH = 0$, the BEPP is exactly the original problem we have solved in the case study. In addition, by setting $MH = 2$, we require the minimum headway between two barges to be 30 minutes. In view of the good sea condition and slow speed of barges in the shelters, this is already too "conservative" for real applications.

We revised the HA to incorporate the queueing and minimum headway requirements (refer to Appendix C.2.3 for details). The revised HA was then used to solve the real case under $MH = 1$ and $MH = 2$. For solving each problem, the algorithm was run 30 time using the settings as presented in Section 4.2. We present the results obtained by the HA for solving the real case under different MH in Appendix D.3. The results demonstrate that the algorithm converges to the same optimal solution in each run for solving the case when $MH = 0$ and $MH = 1$. The objective values obtained by the algorithm for solving the case under $MH = 2$ are also very close to those obtained under $MH = 0$ and $MH = 1$. We have also found that the tug boat and shelter assignment results delivered by the algorithm when solving the case under various MH are also very similar. As for the solution time, solving the case under $MH = 1$ and $MH = 2$ takes longer time than solving the original case. The results indicate that our algorithm is robust against changes in the traffic conditions in the shelters.

5. Discussions

Land reclamation has been widely used around the world as a remedy for insufficient land supply. Various working barges play a critical role in land reclamation projects such that the efficiencies of barges directly affect the process of a land reclamation project. In practice, hiring a working barge is expensive and any construction party of a land reclamation project intends to maximize the utilization of barges as much as possible. Barges are extremely vulnerable to bad weather in the sea and must be evacuated to shelters when facing an approaching storm. Therefore, the BEPP should be considered by the construction parties of many land reclamation projects. However, evacuating barges is not an easy task, as it involves the coordination of barges, tug boats, channels, and shelters. We solve this important yet challenging problem by formulating it as an MIP model, analyzing the features of the problem, and developing a tailored heuristic algorithm.

We test the performance of the algorithm on a number of instances with different parameter settings. The results demonstrate that our algorithm obtains near-optimal solutions when solving problems with small scales and that it beats similar heuristics when solving problems with large scales. We also use the algorithm to solve a real case and the result is better than an evacuation plan generated manually. We have also presented the model, the algorithm, and the results to HKIA, and they agreed with the performance of the algorithm. The model and the algorithm provide references to HKIA in current evacuations. This indicates that our algorithm can provide high-quality evacuation plans to the BEPPs arising in different scenarios.

Our algorithm generates a plan that enables the evacuation to start as late as possible. Using such a plan, the construction party is able to minimize the non-working time of barges in a storm. For the construction party, a late evacuation time means that all barges can work as normal in a longer time. This contributes to lower barge-hiring cost and an earlier completion time of the project.

In the current BEPP, the objective is to maximize the start time. However, it is mentionable that with minor adaptations, the model and the algorithm can also solve the BEPP that aims to minimize the evacuation duration under a given start time. This feature is preferable when the evacuation time window (i.e., the gap between the time a storm is forecast and its arrival time) is short or when safety is the most dominant consideration. In addition to the BEPP, another importation problem is how to move the barges from the shelters back to the project site after the storm. The model and algorithm developed for the BEPP provide references for solving this problem since many similarities are shared by the two problems.

The study is based on the practical problem faced by HKIA, but the proposed model and algorithm are general such that they can be used to solve the BEPPs whose structures are different from the one faced by HKIA. For example, in the BEPP of HKIA, all barges are evacuated in three procedures (i.e., in-site tugging, open-sea tugging, and in-shelter tugging). Now consider the scenarios in which in-site tugging, in-shelter tugging, or both of them are unnecessary. Such scenarios are possible, for example, when the project site is directly connected to the open sea and barges do not have to travel through the channels to get to the open sea, and/or shelters are all of type II. To handle the BEPPs with such structures, one can set the numbers of tug boats and the times required by a barge in the in-site tugging and/or the in-shelter tugging to be zero in the model and in the algorithm.

In the BEPP, all the parameters including the arrival time of the storm and the traveling times between the project site and the shelters are assumed to be deterministic. However, in practice, the evacuation may be affected by uncertainties such that both the arrival time and the traveling times are random. One approach to handle the uncertainties is to set all these parameters in a conservative manner and thus obtain a robust evacuation plan (this is also what we did when solving the case faced by HKIA). This approach is acceptable when high-quality estimations of these parameters can be made. However, when these parameters cannot be accurately estimated or can only be estimated with large variances, using this approach may lead to suboptimal evacuation plans. How to handle uncertainties in different BEPPs is, therefore, an interesting topic for future studies.

6. Conclusion

This paper addresses the BEPP that arises in a practical land reclamation project. We first propose a nonlinear MIP model for the considered problem, and then convert the model into a linear one. We also demonstrate that the general BEPP is strongly NP-hard. To solve the problem, a tailored heuristic algorithm is developed based on the special features of the problem. Extensive numerical experiments are performed and the results demonstrate the algorithm outperforms other solution methods for solving the BEPP with different sizes. We also apply the algorithm to solve a practical problem faced by HKIA, which further demonstrates the efficacy and efficiency of the algorithm. For future studies, a promising topic is to identify any possibilities for further improving the performance of the current algorithm, or for developing more advanced algorithms. As we discussed above, it is also interesting to explore approaches that can solve the BEPP under uncertainties.

Acknowledgments

The authors would like to thank the authority of HKIA for its support in this work. The authors are grateful to the three reviewers for helpful comments

References

Bayram, V., 2016. Optimization models for large scale network evacuation planning and management: a literature review. Surveys in Operations Research and Management Science 21, 63–84.

Ben-Daya, M., Al-Fawzan, M., 1998. A tabu search approach for the flow shop scheduling problem. European Journal of Operational Research 109, 88–95.
Chen, B., Glass, C.A., Potts, C.N., Strusevich, V.A., 1996. A new heuristic for three-machine flow shop scheduling. Operations Research 44, 891–898.
Chen, J.H., Lee, D.H., Cao, J.X., 2011. Heuristics for quay crane scheduling at indented berth. Transportation Research Part E: Logistics and Transportation Review 47, 1005–1020.
Corry, P., Bierwirth, C., 2019. The berth allocation problem with channel restrictions. Transportation Science 53, 708–727.
Dana, G.C., 2009. Floating crane barge Matador 1 under tow. https://www.flickr.com/photos/15610776@N04/6159715160/in/photostream/. Accessed September 9, 2019.
Dixon, M.J., Thompson, G.M., 2016. Bundling and scheduling service packages with customer behavior: model and heuristic. Production and Operations Management 25, 36–55.
Du, J., Leung, J.Y.T., 1989. Complexity of scheduling parallel task systems. SIAM Journal on Discrete Mathematics 2, 473–483.
Fry, J., Binner, J.M., 2016. Elementary modelling and behavioural analysis for emergency evacuations using social media. European Journal of Operational Research 249, 1014–1023.
Garey, M.R., Johnson, D.S., Sethi, R., 1976. The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research 1, 117–129.
Glaser, R., Haberzettl, P., Walsh, R., 1991. Land reclamation in Singapore, Hong Kong and Macau. GeoJournal 24, 365–373.
Glover, F., 1989. Tabu search-part i. ORSA Journal on Computing 1, 190–206.
Hoeksema, R.J., 2007. Three stages in the history of land reclamation in the Netherlands. Irrigation and Drainage 56, 113–126.
Hong Kong Government, 2018. The Chief Executive's 2018 Policy Address. https://www.policyaddress.gov.hk/2018/eng/pdf/PA2018.pdf. Accessed September 9, 2019.
Hong Kong International Airport, 2018. Third runway system–project overview. https://www.threerunwaysystem.com/en/overview/project-overview. Accessed September 9, 2019.
Japan Property Central, 2018. Reclaimed land in Japan. http://japanpropertycentral.com/real-estate-faq/reclaimed-land-in-japan. Accessed September 9, 2019.
Karabuk, S., Manzour, H., 2019. A multi-stage stochastic program for evacuation management under tornado track uncertainty. Transportation Research Part E: Logistics and Transportation Review 124, 128–151.
Lai, D.S., Demirag, O.C., Leung, J.M., 2016. A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph. Transportation Research Part E: Logistics and Transportation Review 86, 32–52.
Leung, J.Y., 2004. Handbook of scheduling: algorithms, models, and performance analysis. CRC press.
Li, S., Jia, S., 2019. The seaport traffic scheduling problem: Formulations and a column-row generation algorithm. Transportation Research Part B: Methodological 128, 158–184.
Martín-Antón, M., Negro, V., del Campo, J.M., López-Gutiérrez, J.S., Esteban, M.D., 2016. Review of coastal land reclamation situation in the world. Journal of Coastal Research 75, 667–671.
Murray-Tuite, P., Wolshon, B., 2013. Evacuation transportation modeling: an overview of research, development, and practice. Transportation Research Part C: Emerging Technologies 27, 25–45.
Ng, M., Diaz, R., Behr, J., 2015. Departure time choice behavior for hurricane evacuation planning: the case of the understudied medically fragile population. Transportation Research Part E: Logistics and Transportation Review 77, 215–226.
OSPAR Commission, 2008. Assessment of the environmental impact of land reclamation. http://qsr2010.ospar.org/media/assessments/p00368_Land_Reclamation.pdf. Accessed September 9, 2019.
Pitana, T., Kobayashi, E., 2009. Optimization of ship evacuation procedures as part of tsunami preparation. Journal of Simulation 3, 235–247.
Schutten, J., 1996. List scheduling revisited. Operations Research Letters 18, 167–170.
Scott, K., 2018. Monaco's 2.3bn dollar project to expand into Mediterranean Sea. https://edition.cnn.com/style/article/monaco-extension-sea/index.html. Accessed September 9, 2019.
Son, S., Wang, M., 2009. Environmental responses to a land reclamation project in South Korea. Eos, Transactions American Geophysical Union 90, 398–399.
Stepanov, A., Smith, J.M., 2009. Multi-objective evacuation routing in transportation networks. European Journal of Operational Research 198, 435–446.

Suzuki, T., 2003. Economic and geographic backgrounds of land reclamation in Japanese ports. Marine Pollution Bulletin 47, 226–229.

Tierney, K., Áskelsdóttir, B., Jensen, R.M., Pisinger, D., 2014. Solving the liner shipping fleet repositioning problem with cargo flows. Transportation Science 49, 652–674.

T'kindt, V., Billaut, J.C., 2006. Multicriteria scheduling: theory, models and algorithms. Springer Science & Business Media.

Wang, W., Liu, H., Li, Y., Su, J., 2014. Development and management of land reclamation in China. Ocean & Coastal Management 102, 415–425.

Wu, L., Wang, S., 2018. Exact and heuristic methods to solve the parallel machine scheduling problem with multi-processor tasks. International Journal of Production Economics 201, 26–40.

Xie, C., Lin, D.Y., Waller, S.T., 2010. A dynamic evacuation network optimization problem with lane reversal and crossing elimination strategies. Transportation Research Part E: Logistics and Transportation Review 46, 295–316.

Yi, W., Nozick, L., Davidson, R., Blanton, B., Colle, B., 2017. Optimization of the issuance of evacuation orders under evolving hurricane conditions. Transportation Research Part B: Methodological 95, 285–304.

Zhao, H., Niu, C., Bai, X., Wang, H., 2017. A nonlinear optimization model for fishing vessel evacuation during typhoon emergencies. Human and Ecological Risk Assessment: an International Journal 23, 457–473.

Appendix A. Model Linearization

We linearize the nonlinear model proposed in Section 3.1.3 by introducing additional decision variables ϖ_{kt}^{i}, ϑ_{gt}^{i}, and φ_{gt}^{i} and replace Constraints (12), (13), (14), (18), and (21) with the following constraints:

$$\varpi_{kt}^{i} \leq \nu_{ki}, \quad \forall k \in \Psi, \forall i \in \Omega, \forall t \in T, \tag{A.1}$$

$$\varpi_{kt}^{i} \leq x_{it}, \quad \forall k \in \Psi, \forall i \in \Omega, \forall t \in T, \tag{A.2}$$

$$\varpi_{kt}^{i} \geq \nu_{ki} + x_{it} - 1, \quad \forall k \in \Psi, \forall i \in \Omega, \forall t \in T, \tag{A.3}$$

$$\varpi_{kt}^{i} \in \{0, 1\}, \quad \forall k \in \Psi, \forall i \in \Omega, \forall t \in T, \tag{A.4}$$

$$\vartheta_{gt}^{i} \leq \mu_{gi}, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.5}$$

$$\vartheta_{gt}^{i} \leq y_{it}, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.6}$$

$$\vartheta_{gt}^{i} \geq \mu_{gi} + y_{it} - 1, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.7}$$

$$\vartheta_{gt}^{i} \in \{0, 1\}, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.8}$$

$$\varphi_{gt}^{i} \leq \mu_{gi}, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.9}$$

$$\varphi_{gt}^{i} \leq z_{it}, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.10}$$

$$\varphi_{gt}^{i} \geq \mu_{gi} + z_{it} - 1, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.11}$$

$$\varphi_{gt}^{i} \in \{0, 1\}, \quad \forall g \in \Theta, \forall i \in \Omega, \forall t \in T, \tag{A.12}$$

$$\varpi_{kt}^{i} = 0, \quad \forall t \in T \setminus \Delta_{ki}, \quad \forall i \in \Omega, \forall k \in \Psi, \tag{A.13}$$

$$\sum_{k \in \Psi} \sum_{i \in \Omega} \sum_{t=\max\{0, t_1 - a_{ki}^{1} - b_{k}^{1} + 1\}}^{t_1} n_i^1 \varpi_{kt}^{i} \leq \sum_{h \in \Phi} \gamma_h^1, \quad \forall t_1 \in T, \tag{A.14}$$

$$\sum_{i \in \Omega} \sum_{t=\max\{0, t_1 - s_i + 1\}}^{t_1} \varpi_{kt}^{i} \leq 1, \quad \forall k \in \Psi, \forall t_1 \in T, \tag{A.15}$$

$$\sum_{g\in\Theta}\sum_{i\in\Omega}\sum_{t=\max\{0,t_1-a^2_{gi}-b^2_g+1\}}^{t_1} n^2_i\vartheta^i_{gt} \le \gamma^2_1,\ \ \forall t_1\in T, \tag{A.16}$$

$$\sum_{i\in\Omega}\sum_{t=\max\{0,t_1-a^3_{gi}-b^3_g+1\}}^{t_1} n^3_i\varphi^i_{gt} \le \sum_{h\in\Phi}\gamma^3_{gh},\ \ \forall g\in\Theta, \forall t_1\in T. \tag{A.17}$$

Constraints (A.1)–(A.4) indicate that $\varpi^i_{kt}=1$ if and only if $\nu_{ki}=1$ and $x_{it}=1$ (i.e., barge i starts sailing in channel k at time point t). Similarly, Constraints (A.5)–(A.8) indicate that $\vartheta^i_{gt}=1$ if and only if $\mu_{gi}=1$ and $y_{it}=1$ (i.e., barge i is assigned to shelter g and starts its open-sea tugging at time point t), and Constraints (A.9)–(A.12) indicate that $\varphi^i_{gt}=1$ if and only if $\nu_{gi}=1$ and $z_{it}=1$ (i.e., barge i is assigned to shelter g and starts berthing into the shelter at time point t). Finally, Constraints (A.13), (A.14), (A.15), (A.16), and (A.17) are the linear versions of Constraints (12), (13), (14), (18), and (21), respectively.

Appendix B. Proof of the Strong NP-hardness

The mathematical proof for Theorem 1 is as follows.

Proof. We transform the Three-Machine Flow Shop Scheduling Problem (3M-FSP) to the decision version of the BEPP. The 3M-FSP can be stated as follows. There are a set Ω of jobs to be processed on three machines (M_1, M_2, and M_3). Each job has to be processed on M_1, then on M_2, and lastly on M_3. The processing time $p^s_i>0$ of each job $i\in\Omega$ on machine M_s ($s=1,2,3$) is given. Preemption is not allowed. Each machine processes at most one job at a time, and each job is processed on at most one machine at a time. The 3M-FSP asks whether there is a processing schedule of jobs denoted by $\mathcal{S}$ such that the makespan (C_{max}) for this problem is no larger than a constant λ.

Given an arbitrary instance of 3M-FSP, we construct a corresponding instance of the BEPP as follows.

There is a set Ω of barges (i's) that have to be evacuated before the deadline D. Each barge needs to be evacuated in 3 procedures (s's), including in-site tugging ($s=1$), open-sea tugging ($s=2$) and in-shelter tugging ($s=3$). In this instance, there is only one available shelter (of Type I). Specifically, we set other parameters as follows (for simplicity, the subscripts for shelters are removed from the parameters).

$$C^{\rm s}=\sum_{i\in\Omega}p_i, \tag{B.1}$$

$$C^{\rm n}=|\Omega|, \tag{B.2}$$

$$f_i=1,\ \ \forall i\in\Omega, \tag{B.3}$$

$$|\Psi|=|\Omega|, \tag{B.4}$$

$$\Gamma_k = \varnothing, \ \ \forall k \in \Psi, \tag{B.5}$$

$$\Delta_{ki} = T, \ \ \forall k \in \Psi, \forall i \in \Omega, \tag{B.6}$$

$$s_i = 0, \ \ \forall i \in \Omega, \tag{B.7}$$

$$a_{ki}^1 = a_i^1, \ \ \forall k \in \Psi, \tag{B.8}$$

$$a_i^s = p_i^s, \ \ \forall i \in \Omega, \forall s \in \{1,2,3\}, \tag{B.9}$$

$$b_k^1 = b^1, \ \ \forall k \in \Psi, \tag{B.10}$$

$$b^s = 0, \ \ \forall s \in \{1,2,3\}, \tag{B.11}$$

$$n_i^s = 1, \ \ \forall i \in \Omega, \forall s \in \{1,2,3\}, \tag{B.12}$$

$$N_1 = 3, \tag{B.13}$$

$$N_2 = 0, \tag{B.14}$$

$$\overline{\lambda} = D - \lambda. \tag{B.15}$$

Clearly, this transformation can be conducted in polynomial time. We will further show that the reduction is pseudo-polynomial by showing that there exists a feasible solution to the constructed instance of BEPP with the starting time of the evacuation $\alpha \geq \overline{\lambda}$ if and only if the answer to the 3M-FSP is "yes". For more details of a pseudo-polynomial reduction, refer to Leung (2004) and T'kindt and Billaut (2006). Suppose the answer to the 3M-FSP is "yes". Let ζ_i^s be the start time of processing job i on machine M_s in schedule $\mathcal{S}$. It is easy to infer that in $\mathcal{S}$ all jobs $i \in \Omega$ are processed by M_s $(s = 1, 2, 3)$ in the same sequence. Let i_n $(n = 1, 2, ..., |\Omega|)$ denote the nth processed job on the machines, then the following three properties must hold for $\mathcal{S}$: (i) $\zeta_{i_{n+1}}^s \geq \zeta_{i_n}^s + p_{i_n}^s, n = 1, 2, ..., |\Omega| - 1, s = 1, 2, 3$, (ii) $\zeta_{i_n}^{s+1} \geq \zeta_{i_n}^s + p_{i_n}^s, n = 1, 2, ..., |\Omega|, s = 1, 2$, and (iii) $\lambda \geq \zeta_{i_{|\Omega|}}^3 + p_{i_{|\Omega|}}^3$.

Then consider the following solution to the constructed instance of the BEPP: (i) assign all barges to shelter G, (ii) assign one HT to each of the three procedures, (iii) evacuate barge i through channel $k = i$ and (iv) for each $i \in \Omega$, barge i starts its evacuation procedure s at time ζ_i^s. Equation (B.1)–(B.3) indicate that every barge can moor into shelter G and the shelter has sufficient capacities to harbor all barges. Equation (B.4) guarantees the feasibility of the channel assignment. Besides, Equation (B.12) ensures that all barges can be evacuated by assigning one tug boat to each procedure. Equation (B.13) indicates that there are sufficient HTs to be assigned to each procedure. Hence, to show the feasibility of the solution, it is sufficient to show that at time ζ_i^s, barge i can start its evacuation in procedure s, and that $\alpha \geq D - \lambda$. We show these as follows.

We first check the feasibility of tugging barges according to $\mathcal{S}$ for the procedure 1 (i.e., the in-site tugging procedure). Equations (B.5) and (B.6) indicate that barges can be tugged into the channels (i.e., start procedure $s = 1$) at any time (tidal conditions are always suitable and there always exists a channel that is free of traffic) and in any sequence (without precedence constraints). It is obvious that barge i_1 can start the in-site tugging at time $\zeta_{i_1}^1$. Now suppose that barge i_n ($n = 1, 2, ..., |\Omega| - 1$) starts the in-site tugging at time $\zeta_{i_n}^1$. Then i_{n+1} can start its in-site tugging as early as $\zeta_{i_n}^1 + a_{i_n}^1 + b^1$. Considering that $a_{i_n}^1 = p_{i_n}^1$, $b^1 = 0$, and $\zeta_{i_{n+1}}^1 \geq \zeta_{i_n}^1 + p_{i_n}^1, n = 1, 2, ..., |\Omega| - 1$, it is feasible to start the in-site tugging procedure of barge i_{n+1} at time $\zeta_{i_{n+1}}^1$. Therefore, by induction, the feasibility of $\mathcal{S}$ for tugging barges in the in-site tugging procedure is proved. Then, look at the second procedure (i.e., the open-sea tugging). It is easy to see that barge i_1 can start the open-sea tugging at time $\zeta_{i_1}^2$ since $\zeta_{i_1}^2 \geq \zeta_{i_1}^1 + p_{i_1}^1 = \zeta_{i_1}^1 + a_i^1$. Suppose barge i_n ($n = 1, 2, ..., |\Omega| - 1$) starts the open-sea tugging at time $\zeta_{i_n}^2$. Then consider that (i) $\zeta_{i_{n+1}}^2 \geq \zeta_{i_{n+1}}^1 + p_{i_{n+1}}^1 = \zeta_{i_{n+1}}^1 + a_{i_{n+1}}^1$ where $\zeta_{i_{n+1}}^1$ is the time barge i_{n+1} starts its in-site tugging and that (ii) $\zeta_{i_{n+1}}^2 \geq \zeta_{i_n}^2 + p_{i_n}^2 = \zeta_{i_n}^2 + a_{i_n}^2 + b^2$ ($a_{i_n}^2 = p_{i_n}^2$ and $b^2 = 0$). It follows that barge i_{n+1} can start the open-sea tugging at time $\zeta_{i_{n+1}}^2$. Therefore, the feasibility of $\mathcal{S}$ for tugging barges in the open-sea tugging procedure can also be proved by induction. Following the same procedure we can also verify that barges can be tugged according to $\mathcal{S}$ in the in-shelter tugging procedure. Finally, in this constructed solution, the last evacuated barge (i.e., $i_{|\Omega|}$) starts berthing into shelter G at time $\zeta_{i_{|\Omega|}}^3$, leading to the makespan equal to $\zeta_{i_{|\Omega|}}^3 + a_{i_{|\Omega|}}^3 = \zeta_{i_{|\Omega|}}^3 + p_{i_{|\Omega|}}^3 \leq \lambda$. Therefore, we have $\alpha \geq D - \lambda = \overline{\lambda}$.

Conversely, suppose that there exists a feasible solution to the constructed instance of the BEPP such that $\alpha \geq D - \lambda$. Firstly, as there is only one shelter that is able to harbor all barges, in any feasible solution to the instance, all barges should be assigned to the shelter. Secondly, considering that there are only 3 tug boats (HTs) and 3 evacuation procedures, in any feasible solution to the BEPP instance, exactly one barge should be assigned to one procedure. Finally, denote the schedule of barge evacuation procedures by $\mathcal{E}$, and let ε_i^s denote the time when barge $i \in \Omega$ starts procedure $s = 1, 2, 3$. It is easy to know that all barges $i \in \Omega$ are evacuated in the same sequence in different procedures. Let i_n ($n = 1, 2, ..., |\Omega|$) be the nth evacuated barge, then the following three properties must hold for $\mathcal{E}$: (i) $\varepsilon_{i_{n+1}}^s \geq \varepsilon_{i_n}^s + a_{i_n}^s + b^s, n = 1, 2, ..., |\Omega| - 1, s = 1, 2, 3$, (ii) $\varepsilon_{i_n}^{s+1} \geq \varepsilon_{i_n}^s + a_{i_n}^s, n = 1, 2, ..., |\Omega|, s = 1, 2$, and (iii) $\lambda \geq \varepsilon_{i_{|\Omega|}}^3 + a_{i_{|\Omega|}}^3$. Considering $a_{i_n}^s = p_{i_n}^s$ and $b^s = 0$ for $s = 1, 2, 3$, the properties are equivalent to: (i) $\varepsilon_{i_{n+1}}^s \geq \varepsilon_{i_n}^s + p_{i_n}^s, n = 1, 2, ..., |\Omega| - 1, s = 1, 2, 3$, (ii) $\varepsilon_{i_n}^{s+1} \geq \varepsilon_{i_n}^s + p_{i_n}^s, n = 1, 2, ..., |\Omega|, s = 1, 2$, and (iii) $\lambda \geq \varepsilon_{i_{|\Omega|}}^3 + p_{i_{|\Omega|}}^3$.

Obviously, scheduling jobs in a fashion such that job i starts being processed on machine M_s at time ε_i^s generates a feasible schedule with $C_{max} \leq \lambda$ to the 3M-FSP. This completes the proof. □

Appendix C. Algorithm Details

We present the details of the heuristic algorithm in this appendix.

Appendix C.1. Pseudo-code of the Algorithm for the Shelter Assignment Procedure

Algorithm 1 The shelter assignment procedure.

Input: The berthing space (p_i) of each barge i, the capacities (C_g^{s} and C_g^{n}) of each shelter g, the time (a_{gi}^2) of tugging each barge i from the project site to each shelter g, and the shelter-barge compatibility index f_{gi} of each pair (g, i);

Output: The assigned shelter (as_i) for each barge i and set of barges Ω_g accommodated by shelter g;

Initialize $\Omega_g = \varnothing$, $\forall g \in \Theta$;
Initialize the set of barges that have not been assigned to any shelter: $\Omega' = \Omega$;
Initialize the available capacities of the shelters: $lc_g^{\mathrm{s}} = C_g^{\mathrm{s}}$, and $lc_g^{\mathrm{n}} = C_g^{\mathrm{n}}$, $\forall g \in \Theta$;
while $\Omega' \neq \varnothing$ **do**
    $I = \arg\max_{i \in \Omega'} p_i$;
    $\Theta' = \{g | lc_g^{\mathrm{s}} \geq p_I, lc_g^{\mathrm{n}} \geq 1, f_{gi} = 1, g \in \Theta\}$;
    $G = \arg\min_{g \in \Theta'} a_{gI}^2$;
    $as_I = G$;
    $\Omega_g = \Omega_g \bigcup \{I\}$;
    $lc_g^{\mathrm{s}} = lc_g^{\mathrm{s}} - p_I$;
    $lc_g^{\mathrm{n}} = lc_g^{\mathrm{n}} - 1$;
    $\Omega' = \Omega' \setminus \{I\}$;
end while

Appendix C.2. Details of the Heuristic for Solving the T-BEPP and the R-BEPP

Appendix C.2.1. A Tabu Search Heuristic for Tug Boat Assignment

TS is a local-search-based meta-heuristic designed to find near-optimal solutions for combinatorial optimization problems. The method was originally proposed by Glover (1989) and has been widely applied in solving practical assignment and scheduling problems (e.g., Chen et al., 2011 and Lai et al., 2016). In this step, we chose TS because (i) the problem is of high complexity, (ii) the neighborhood of an assignment problem is relatively narrow and (iii) similar solutions are more likely to be generated (in comparison with a sequencing problem).

In the TS, the solutions can be presented by a $2 \times (2 + N)$ matrix (denoted by Ξ), where N equals the number of shelters selected to harbor barges in the shelter assignment step (refer to Section 3.3.3) and the derived vectors Ξ_1 and Ξ_2 demonstrate the assignments of HTs and LTs, respectively. In particular, Ξ_1^1 (resp. Ξ_2^1) and Ξ_1^2 (resp. Ξ_2^2) are the numbers of HTs (resp. LTs) assigned to the in-site tugging and open-sea tugging procedures, respectively, and Ξ_1^{2+n} (resp. Ξ_2^{2+n}) where $n \in \{1, 2, ..., N\}$ denotes the number of HTs (resp. LTs) assigned to the nth selected shelter.

Tug Boat Assignment Initialization. TS is by nature a local-search-based optimization algorithm, and in each iteration, it searches in the neighborhood of the current solution to find a new solution which will then replace the current one. In the TS, we use the initial tug boat assignment (denoted by $\widetilde{\Xi}$) obtained by the following procedure to provide a starting point for the local search procedure when the TS is run for the first time (i.e., in the "Initial Makespan Calculation" step of the first stage of the solution algorithm; see Figure 7). In the procedure, we try to assign tug boats in a balanced manner, and the detailed steps are listed as follows:

Step 1. Assign all the HTs to the sea-going tugging procedure.

Step 2. Assign half of the LTs to the in-site tugging procedure.

Step 3. Assign, from the remaining half of the LTs, $\max_{i \in \Omega_g}\{n_{gi}^3\}$ LTs to each shelter g which has been selected to accommodate barges.

Step 4. For LTs that have not been assigned in Steps 2 and 3, assign them to each shelter in proportion to $\sum_{i \in \Omega_g} n_{gi}^3$.

In the subsequent runs (i.e., in the "Makespan Check" step of the second stage of the solution algorithm; see Figure 7), the TS starts from the best assignment pattern obtained in the previous runs.

Neighborhood Construction. To construct the neighborhood (denoted by $NB(\Xi)$) for the current solution Ξ, we start by identifying boundaries for Ξ_c^d's where $c \in \{1, 2\}$ and $d \in \{1, 2, ..., 2 + N\}$. To begin with, the following conditions provide valid lower and upper bounds for Ξ_c^d's:

$$\sum_{c \in \{1,2\}} \Xi_c^1 \geq \max_{i \in \Omega} n_i^1, \tag{C.1}$$

$$\Xi_1^2 \geq \max_{i \in \Omega} n_i^2, \tag{C.2}$$

$$\Xi_2^2 = 0, \tag{C.3}$$

$$\sum_{c\in\{1,2\}} \Xi_c^{2+n} \geq \max_{i\in\Omega_{g_n}} n_{g_n i}^3, \tag{C.4}$$

$$\sum_{c\in\{1,2\}} \Xi_c^{2+n} \leq \sum_{i\in\Omega_{g_n}} n_{g_n i}^3, \tag{C.5}$$

where g_n stands for the nth used shelter and Ω_{g_n} denotes the set of barges assigned to this shelter.

To construct the $NB(\Xi)$ for Ξ, we generate new solutions by changing the assignment of HTs and LTs in Ξ (one tug boat at a time) and all the new solutions that satisfy conditions (C.1)–(C.5) will be added into $NB(\Xi)$.

Search Procedure. Before proposing the search procedure of the TS, we convert a solution Ξ into a set $[\Upsilon(\Xi)]$ of N tuples. In particular, tuple $\mathbb{P}_n = (TN_1, TN_2, TN_3^n)$ where $n \in \{1, 2, ..., N\}$ corresponds to an assignment pattern with TN_1, TN_2 and TN_3^n tug boats assigned to in-site tugging, open-sea tugging, and the in-shelter tugging of the nth used shelter, respectively. Obviously, any transition between Ξ and a new solution Ξ' in $NB(\Xi)$ will correspondingly get $\Upsilon(\Xi)$ replaced by $\Upsilon(\Xi')$ where some tuples are removed and some new ones are added.

In addition, we introduce the following notation. Let $C_{\max}(\Xi)$ denote the makespan under the current solution Ξ ($C_{\max}(\Xi)$ is obtained by the procedures given in Section Appendix C.2.2 and Section Appendix C.2.3). In addition, let PM be the punishment multiplier for generating repeating tuples when a new solution is created to replace the current one and TL be the tabu length. The iterations within which tuple $\mathbb{P}_n$ is in tabu is denoted by $\pi(\mathbb{P}_n)$. The aspiration level of tuple $\mathbb{P}_n$ is denoted by $\lambda(\mathbb{P}_n)$. Further, the frequency of tuple $\mathbb{P}_n$ being generated in solutions is denoted by $\mu(\mathbb{P}_n)$.

The search procedure follows the rules shown below:

- Some transitions from Ξ to $\Xi' \in NB(\Xi)$ may generate $\mathbb{P}_n$'s that are in tabu. These transitions should be forbidden if all the newly generated $\mathbb{P}_n$'s are in tabu. However, if Ξ' yields a makespan that is smaller than the aspiration level $\lambda(\mathbb{P}_n)$ of at least one newly generated $\mathbb{P}_n$, Ξ' should be considered as a valid neighboring solution for Ξ, no matter whether there are newly generated $\mathbb{P}_n$'s that are held in tabu or not;

- In the algorithm, each Ξ' is evaluated in terms of both the derived makespan and the level of originality. Therefore, a Ξ' will be punished if some $\mathbb{P}_n$'s that are newly added into $\Upsilon(\Xi)$ have been generated before. However, Ξ' with $C_{\max}(\Xi')$ smaller than the incumbent $C_{\max}(\Xi)$ is exempted from such punishment;

- Ξ' with the smallest $v(\Xi')$ (denoted by $\widehat{\Xi}$) is selected to replace Ξ in the next iteration ($v(\Xi')$'s are calculated by adding the punishments (if any) for generating repeating $\mathbb{P}_n$'s to the makespans $C_{\max}(\Xi')$'s). $\mathbb{P}_n$'s in $\Upsilon(\Xi) \setminus \Upsilon(\widehat{\Xi})$ (i.e., $\mathbb{P}_n$'s that are removed from the current solution) are held in tabu for the next TL iterations.

We are now ready to introduce the search procedure of the TS, which is given in Algorithm 2.

Algorithm 2 The search procedure of the TS.

Input: Initial solution: $\widetilde{\boldsymbol{\Xi}}$ and controlling parameters: PM, TL;
Output: Best makespan $C^*_{\max}$ and best solution $\boldsymbol{\Xi}^*$;

$\pi(\mathbb{P}_n) = 0$, $\lambda(\mathbb{P}_n) = M$ (M is a large constant), and $\mu(\mathbb{P}_n) = 0$, $\forall \mathbb{P}_n$, $\forall n \in \{1, 2, ..., N\}$;
$\boldsymbol{\Xi}$=$\widetilde{\boldsymbol{\Xi}}$;
$\boldsymbol{\Xi}^* = \boldsymbol{\Xi}$, $C^*_{\max}$=$C_{\max}(\boldsymbol{\Xi})$;
$\lambda(\mathbb{P}_n) = C^*_{\max}, \forall \mathbb{P}_n \in \Upsilon(\boldsymbol{\Xi})$;
for $\kappa = \{1, ..., \kappa_{\max}\}$ **do**
  Construct neighbourhood $NB(\boldsymbol{\Xi})$;
  Initialize the set of valid candidates $\overline{NB}(\boldsymbol{\Xi})$ in $NB(\boldsymbol{\Xi})$ as $\overline{NB}(\boldsymbol{\Xi}) = \varnothing$;
  for $\boldsymbol{\Xi}' \in NB(\boldsymbol{\Xi})$ **do**
    Initialize validness VA of $\boldsymbol{\Xi}'$ as $VA = 0$;
    for $\mathbb{P}_n \in \Upsilon(\boldsymbol{\Xi}') \setminus \Upsilon(\boldsymbol{\Xi})$ **do**
      if $\pi(\mathbb{P}_n) < \kappa$ or $C_{\max}(\boldsymbol{\Xi}') < \lambda(\mathbb{P}_n)$ **then**
        $VA = 1$;
        Break the For-loop;
      end if
    end for
    if $VA = 1$ **then**
      $\overline{NB}(\boldsymbol{\Xi}) = \overline{NB}(\boldsymbol{\Xi}) \bigcup \boldsymbol{\Xi}'$;
    end if
  end for
  for $\boldsymbol{\Xi}' \in \overline{NB}(\boldsymbol{\Xi})$ **do**
    if $C_{\max}(\boldsymbol{\Xi}') < C_{\max}(\boldsymbol{\Xi})$ **then**
      $v(\boldsymbol{\Xi}') = C_{\max}(\boldsymbol{\Xi}')$;
    else
      $v(\boldsymbol{\Xi}') = C_{\max}(\boldsymbol{\Xi}') + PM \sum_{\mathbb{P}_n \in \Upsilon(\boldsymbol{\Xi}') \setminus \Upsilon(\boldsymbol{\Xi})} \mu(\mathbb{P}_n)$;
    end if
  end for
  if $\overline{NB}(\boldsymbol{\Xi}) \neq \varnothing$ **then**
    $\widehat{\boldsymbol{\Xi}} = \arg\min_{\boldsymbol{\Xi}' \in \overline{NB}(\boldsymbol{\Xi})} v(\boldsymbol{\Xi}')$;
    for $\mathbb{P}_n \in \Upsilon(\widehat{\boldsymbol{\Xi}}) \setminus \Upsilon(\boldsymbol{\Xi})$ **do**
      $\mu(\mathbb{P}_n) = \mu(\mathbb{P}_n) + 1$;
    end for
    for $\mathbb{P}_n \in \Upsilon(\boldsymbol{\Xi}) \setminus \Upsilon(\widehat{\boldsymbol{\Xi}})$ **do**
      $\pi(\mathbb{P}_n) = \kappa + TL$;
    end for
    for $\mathbb{P}_n \in \Upsilon(\widehat{\boldsymbol{\Xi}})$ **do**
      if $C_{\max}(\widehat{\boldsymbol{\Xi}}) < \lambda(\mathbb{P}_n)$ **then**
        $\lambda(\mathbb{P}_n) = C_{\max}(\widehat{\boldsymbol{\Xi}})$;
      end if
    end for
    if $C^*_{\max} > C_{\max}(\widehat{\boldsymbol{\Xi}})$ **then**
      $C^*_{\max} = C_{\max}(\widehat{\boldsymbol{\Xi}})$;
      $\boldsymbol{\Xi}^* = \widehat{\boldsymbol{\Xi}}$;
    end if
    $\boldsymbol{\Xi} = \widehat{\boldsymbol{\Xi}}$;
  end if
end for

Appendix C.2.2. A Simulated Annealing Algorithm for Barge Sequencing

This section introduces the SA for optimizing the evacuation sequence among the barges under a given tug boat assignment pattern delivered by the TS algorithm. SA has been applied to solve many complicated combinatorial optimization problems (Tierney et al., 2014 and Dixon and Thompson, 2016, for instance). We chose SA in this step because of the discrete nature of the solution and the complexity of a solution. Other popular search heuristics (e.g., Tabu Search, Genetic Algorithm) require memory for multiple solutions, which can be very large, especially for a sequencing problem. Like TS, SA starts with an initial solution and then improves the current solution by searching in its neighborhood.

Evacuation Sequence Initialization. To obtain high-quality initial sequences, the following procedures are used to generate the initial solutions. We initialize the barge evacuation sequences for the SA in the first iteration of the TS when solving the T-BEPP and in the first iteration of the first run of the TS when solving the R-BEPP. Note that in other iterations of the TS to solve the T-BEPP or the R-BEPP, the SA starts from the best sequence obtained in the previous runs. The sequence is constructed by adding barges one by one. At each time for sequence extension, we select the barge to extend the sequence in a greedy way. The extension procedure includes two steps. First, we assign each barge i a sequencing weight w_i which is calculated by $w_i = 1000BN_i - 100p_i + D_i$, where BN_i denotes the number of barges blocked by barge i from sailing into the channels, and D_i stands for the draft of the barge. During the sequencing process, in any set Ω_g of shelter g (Ω_g is the set of barges assigned to shelter g), the barge with the largest w_i will be searched first. We set w_i in this manner to evacuate barges that block others from using certain channels first, and to evacuate barges with smaller sizes first. This enables us to make all channels accessible to barges as soon as possible, and evacuating smaller barges first leads to a more flexible usage of tug boats.

When adding a new barge into the sequence, we tend to select the one that generates the least waiting time in the outer anchorages of the shelters (i.e., the time inteval between the arrival of the barge in the outer anchorage of a shelter and the start of the in-shelter tugging procedure). In addition to the waiting time of barges, the delays of tug boats and the balance among workloads of tug boat in shelters are also considered when sequencing the barges. Algorithms 3 and 4 demonstrate the detailed procedure. Additional notation used in the algorithms is introduced in Table C.1.

Table C.1: Additional notation used in Algorithms 3 and 4.

σ	Queue that records the sequence to start evacuating the barges.
dl_i^1	Delay caused by tidal conditions for barge i in the in-site tugging procedure.
dl_i^2	Delay caused by tidal conditions for barge i in the open-sea tugging procedure.
dl_i^3	Waiting time of barge i in the outer anchorages of the shelters.
AB_g	Number of barges that have been added into σ, among those assigned to shelter g.
Ω'_g	Set of barges that are assigned to shelter g and have not been added into σ.
SI_i	Index for barge selection when a partial σ is extended, which is calculated by Equation (C.6).
ts^1	Set of tug boats assigned to the in-site tugging procedure.
ts^2	Set of tug boats assigned to the open-sea tugging procedure.
ts_g^3	Set of tug boats assigned to the in-shelter tugging procedure of shelter g.
tr_h	Ready time of tug boat h to serve barges.
cr_k	Ready time of channel k to evacuate barges.
q_i^1	Earliest time for barge i to start in-site tugging.
q_i^2	Earliest time for barge i to start open-sea tugging.
q_i^3	Earliest time for barge i to start in-shelter tugging.
M	A large constant.
$\min[n](S)$	Function that returns the nth smallest value in set S.

SI_i is calculated using the following equation:

$$SI_i = sw_i^1 dl_i^1 + sw_i^2 dl_i^2 + sw_i^3 dl_i^3 + sw_i^4 AB_G, \tag{C.6}$$

where $G = as_i$ is obtained by Algorithm 1, and sw_i^1, sw_i^2, and sw_i^3 are the weights for dl_i^1, dl_i^2, and dl_i^3, respectively. For a given i, we set $sw_i^1 = 1/CN + n_i^1/TN_1$, $sw_i^2 = n_i^2/TN_2$, $sw_i^3 = 1$, and $sw_i^4 = 1$, where CN represents the number of channels, and TN_1 and TN_2 stand for the total number of tug boats assigned to the in-site tugging procedure and the open-sea tugging procedure, respectively. Given a group of candidate barges, the one with the least SI_i will be selected to extend σ. Note that sw_i^1 and sw_i^2 are set equal to the proportion of tug boats or channels that are affected by the delay caused by tidal conditions. Meanwhile, by setting sw_i^3 and sw_i^4 to be 1, which are generally larger than sw_i^1 and sw_i^2, we put more weights on balancing the workload among shelters. Furthermore, when tidal conditions are omitted (i.e., in the T-BEPP), for any i, dl_i^1 and dl_i^2 equal 0, and thus only dl_i^3 and AB_G are considered.

Algorithm 3 The initial sequencing procedure in the SA.

Input: The sequencing weight (w_i) for each barge i, and the set of barges (Ω_g) to be evacuated to each shelter g.

Output: σ;

```
Initialize σ = ∅, tr_h = 0, ∀h ∈ Φ, cr_k = 0, ∀k ∈ Ψ, AB_g = 0 and Ω'_g = Ω_g, ∀g ∈ Θ and set
  Θ' = {g | Ω'_g ≠ ∅, g ∈ Θ};
while Θ' ≠ ∅ do
    Initialize the upper bound for SI_i's: SI‾ = M;
    Update the set of barges that have not been added into σ: Ω' = ⋃_{g∈Θ'} Ω'_g;
    for g ∈ Θ' do
        Ω''_g = Ω'_g;
        while Ω''_g ≠ ∅ do
            I = arg min_{i∈Ω''_g}(w_i);
            Calculate SI_I using Algorithm 4;
            if SI_I < SI‾ then
                SI‾ = SI_I, i* = I;
            end if
            if dl_I^1 = 0 (dl_I^1 is obtained by Algorithm 4) then
                Break the inner While-loop;
            end if
            Ω''_g = Ω''_g \ {I};
        end while
    end for
    σ = σ ∪ {i*};
    G = as_{i*} (as_{i*} is the shelter assigned to harbor barge i*.);
    AB_G = AB_G + 1;
    Ω'_G = Ω'_G \ {i*};
    if Ω'_G = ∅ then
        Θ' = Θ' \ G;
    end if
    Update tr_h and cr_k for evacuating barge i* using List Scheduling;
end while
```

Algorithm 4 The procedure to calculate SI_I.

Input: tr_h, cr_k, AB_g, $G = as_I$ (the shelter assigned to harbor barge I), and Ω' which is obtained from Algorithm 3;

Output: SI_I, and dl_I^1;

Initialize $SI_I = M$, and $q_I^1 = M$;
for $k \in \Psi$ **do**
    if $\bigcup_{j \in \Omega'} \{j | (j, I) \in \Gamma_k\} = \varnothing$ **then**
        if $q_I^1 > cr_k$ **then**
            $q_I^1 = cr_k$;
            $K = k$;
        end if
    end if
end for
if $q_I^1 < M$ **then**
    if $\min[n_I^1]_{h \in ts^1}(tr_h) > q_I^1$ **then**
        $q_I^1 = \min[n_I^1]_{h \in ts^1}(tr_h)$;
    end if
    if $q_I^1 \in \Delta_{KI}$ **then**
        $dl_I^1 = 0$;
    else
        $e_I = \min_{t \in \Delta_{KI} : t \geq q_I^1}(t)$
        $dl_I^1 = e_I - q_I^1$;
        $q_I^1 = e_I$;
    end if
    $q_I^2 = \min[n_I^2]_{h \in ts^2}(tr_h)$;
    if $q_I^2 \geq q_I^1 + a_{KI}^1$ **then**
        $dl_I^2 = 0$;
    else
        $q_I^2 = q_I^1 + a_{KI}^1$;
        if $q_I^2 > q_I^1 + a_{KI}^1 - dl_I^1$ **then**
            $dl_I^2 = q_I^1 + a_{KI}^1 - q_I^2$;
        else
            $dl_I^2 = dl_I^1$;
        end if
    end if
    $q_I^3 = \min[n_{GI}^3]_{h \in ts_G^3}(tr_h)$;
    if $q_I^3 < q_I^2 + a_{GI}^2$ **then**
        $dl_I^3 = 0$;
    else
        $dl_I^3 = q_I^3 - (q_I^2 + a_{GI}^2)$;
    end if
end if
Calculate SI_I using Equation (C.6);

Neighborhood Construction and Search Procedure. The SA seeks to improve the initial solution in a given number of iterations. In each iteration, the algorithm searches in the neighborhood of the incumbent solution and selects one solution from the neighborhood to replace the incumbent. For an incumbent solution, we construct its neighborhood by taking advantage of Observation 2. In particular, we generate the neighborhood of the solution by swapping each barge in the solution with its followers in σ, one by one, until a follower that is assigned to the same shelter with the barge is found. Details for constructing the neighborhood are demonstrated in Algorithm 5, where we let $\sigma = [i_1, i_2, ..., i_{|\Omega|}]$ denote the incumbent solution and as_{i_n} be the assigned shelter for barge i_n.

Algorithm 5 The neighborhood construction procedure for the SA.

Input: The incumbent solution (σ);
Output: Neighborhood of the solution $NB(\sigma)$;

Initialize $NB(\sigma) = \varnothing$;
for $n \in \{1, 2, ..., |\Omega| - 1\}$ **do**
    for $m \in \{i + 1, i + 2, ..., |\Omega|\}$ **do**
        if $as_{i_m} \neq as_{i_n}$ **then**
            Generate a new solution by swapping i_n and i_m, and add it into $NB(\sigma)$;
        else
            Break the inner For-loop;
        end if
    end for
end for

After the neighborhood of the incumbent solution is constructed, we calculate the makespan of each solution using the method proposed in Section Appendix C.2.3. Then, we identify the solution with the minimum makespan (we denote the solution and the corresponding makespan by σ^* and $C_{\max}(\sigma^*)$, respectively), if $C_{\max}(\sigma^*) < C_{\max}(\sigma)$, we replace σ with σ^*, otherwise, we replace σ with σ^* with probability $p = e^{\frac{C_{\max}(\sigma^*) - C_{\max}(\sigma)}{bc\tau}}$, where bc is Boltzmann constant and τ is the current "temperature" of the system. In particular, τ is calculated by $\tau = \tau_0 \rho^{iters}$, where τ_0 is the "starting temperature", $0 < \rho < 1$ is the "temperature reduction rate", and $iters$ is the current number of iterations exacuated in the SA.

Appendix C.2.3. Resource Scheduling

Resource scheduling aims at assigning the channel and the tug boats to serve each barge in each evacuation procedure and deciding the start time of each evacuation procedure of each barge, where the assignment of barges among shelters (see Section 3.3.3), the assignment pattern of tug boats among evacuation procedures and shelters (see Section Appendix C.2.1), and the evacuation sequence of the barges (see Section Appendix C.2.2) are all known. As suggested in Observation 3, we adopt List Scheduling in this step. Note that we can extend the resource scheduling procedure to strictly control congestion in the shelters. In particular, we can take the entrance of a shelter as a resource. Then, we set a minimum headway between two barges that sail consecutively into the shelter when "assigning" the entrance to the barges.

Appendix D. Supplementary Data in the Case Study

We present the supplementary data in the case study in this appendix.

Appendix D.1. Input Data of the Real Case

The input data of the BEPP face by HKIA are listed as follows. To begin with, table D.2 shows the configuration of the barge fleet (including 102 barges). These barges are divided into three groups according to their sizes. Besides, among all barges, there are 90 small draft barges (whose drafts are within $[2, 6]$) and 12 large draft barges (whose drafts are within $(6, 9]$).

Table D.2: Parameters of the barges in the real case.

Size group	Length range(m)	Shelter space[1,2]	Number of barges
Small	[0, 50]	1.0	52
Medium	[50, 75]	1.5	40
Large	[75, 100]	2.0	10

Note[1]: The number of unit sizes a barge occupies when berthing in a shelter.
Note[2]: One unit size is a 50-metre-long equivalent.

The parameters of the shelters are shown in Table D.3. Note that in this table the capacity of a shelter is reported in the format of "total available space (largest number of barges it can harbor)". If there are no particular limitations for the number of barges, we report it by "–". Besides, we report the time taken for the open-sea tugging of barges from the project site to different shelters in columns named "Time 1", and report the sailing time of tug boats from shelters back to the 3RS project site in columns named "Time 2". In addition, for shelters where the "Eastward Evacuation Routes" (i.e., routes without traversing the HZM bridge) are shorter, all barges evacuated to them will use the "Eastward Evacuation Routes" in the open-sea tugging procedure. However, as shown in Figure 2, for shelters where the "Westward Evacuation Routes", (i.e., routes traversing the HZM bridge) are shorter, barges evacuated to them may use different routes. In particular, barges having a height less than 41 meters will use the "Westward Evacuation Routes", while barges higher than 41 meters will use the "Eastward Evacuation Routes", due to the limited vertical clearance of the HZM bridge. Note that for each route of each shelter we set congruent "Time 1"s (resp. "Time 2"s) for all barges (resp. tug boats). In Table D.3, we report the attributes of routes that are only available for barges with a height less than 41 meters in the column named "Route 2", and the attributes of routes that are available for all barges are reported in columns named "Route 1". Of the 102 barges, 24 of them are higher than 41 meters.

Table D.3: Parameters of the shelters in the real case.

Shelter	Type	Capacity (in unit sizes)	Size limit[1]	Route 1		Route 2	
				Time 1 (h)	Time 2 (h)	Time 1 (h)	Time 2 (h)
A	I	60(–)	Large	2.5	0.5	–	–
B	I	20(–)	Large	3.0	1.0	–	–
C	II	6(3)[2]	Large	3.0	1.0	–	–
D	I	15(–)	Large	10.5	3.0	8.5	2.5
E	I	30(–)	Small	6.0	2.0	–	–
F	I	30(–)	Large	2.0	0.5	–	–

Note[1]: The largest size of barges a shelter can harbour.
Note[2]: In Shelter C, each vessel moors using two mooring buoys, regardless of their sizes. In this shelter, there are in total six mooring buoys, and thus at most 3 vessels can berth in it.

A fleet of tug boats is available to serve the barges during the evacuation. The tug fleet is composed of 30 HTs and 20 LTs. Table D.4 demonstrates the parameters of the tug boats in different evacuation procedures.

Table D.4: Parameters of tug boat usage.

Procedure	Barge Size	Type	Number	Time 1 (h)[1,3]	Time 2 (h)[2,3]
In-site tugging	Small	HTs and LTs	1	2.0	0.25
	Medium	HTs and LTs	1		
	Large	HTs and LTs	2		
Open-sea tugging	Small	HTs	1	Refer to Table D.3	
	Medium	HTs	2		
	Large	HTs	3		
In-shelter tugging[4]	Small	HTs and LTs	1	2.5	0.25
	Medium	HTs and LTs	1		
	Large	HTs and LTs	1		

Note[1]: Time for tugging a barge in different evacuation procedures.
Note[2]: Time for a tug boat to return to the site area, outer anchorage of the site, or outer anchorage of the shelter after completing a towage in different evacuation procedures.
Note[3]: "Time 1"s and "Time 2"s are congruent for all barges and tug boats, respectively.
Note[4]: Only for Type I shelters.

There are three channels that connect the project site with the open sea. For barges that sail consecutively in the same channel, the minimum interval between the start times of their in-site tugging procedures is 15 minutes. The tidal condition is shown in Figure D.1, which gives an illustration of feasible time windows for tugging barges with different drafts into the channels on a typical day. In particular, barges with small drafts (i.e., drafts no larger than Draft 1 in this figure) can be tugged into the channels at any time. However, for barges whose drafts are larger than Draft 1, they can only be tugged into the channels within suitable time windows, and the time windows become narrower for barges with larger drafts (e.g., the time windows for barges with Draft 3 are narrower than those with Draft 2). We simulate the tidal condition in the channels using the curve $9 + 3\sin\frac{\pi h}{24}$, where $h = \{1, ..., 96\}$ represents the length (unit times) after zero o'clock in a day, and a unit time equals 15 minutes. For each channel, three or four barges work near the entrance of the channel, and they should be evacuated out of the project site first before other barges sail into the channel.

Appendix D.2. Shelter and Tug Boat Assignment Results in the Real Case

The shelter and tug boat assignment results obtained in one of the runs of the HA are shown in Table D.1 (there are small variations in tug boat assignments among different runs).

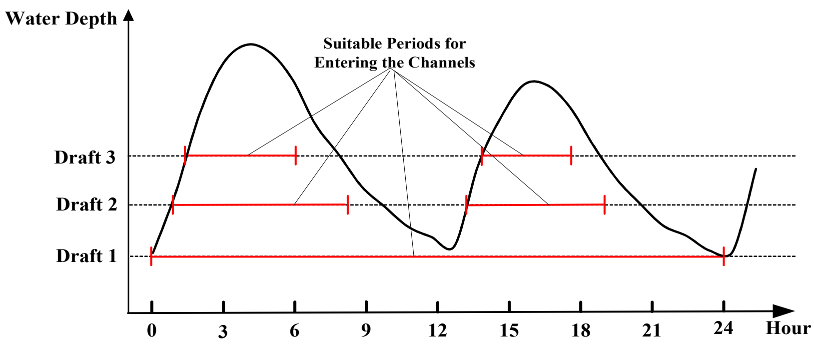

Figure D.1: The tidal condition in the channels in the real case.

Table D.1: The resource assignment result.

Shelter	Accommodated Barges			Assigned Tug boats	
	Small Barges	Medium Barges	Large Barges	LTs	HTs
A	0	26	10	4	0
B	19	0	0	2	1
C[1]	3	0	0	–	–
D	0	0	0	0	0
E	22	0	0	3	0
F	8	14	0	3	0
Total	52	40	10	12	1
Tug boats assigned to the in-site tugging procedure				8	5
Tug boats assigned to the open-sea tugging procedure				0	24

Note[1]: Barges can be berthed into Shelter C directly from the open sea, and no specific tug boats for in-shelter tugging are required.

Appendix D.3. Results of the Real Case under Different Traffic Control in the Shelters

Table D.2 reports the performances of the HA for solving the real case under different settings of the minimum headway between two barges in the shelters. Column 1 presents the value of MW in each instance. Columns 2 to 4 report the average objective value (i.e., α), the average objective value (i.e., α) of the best solution, and the average objective value (i.e., α) of the worst solution in 30 runs, respectively. Column 5 presents the average solution time (in seconds) in the 30 runs.

Table D.2: Solution Results of the Real Case under Different MW.

Headway (MW)	Solution			Time(s)
	Average	Best	Worst	
0	118.00	118	118	693.33
1	118.00	118	118	857.63
2	115.40	117	115	1104.90