Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 246978, 22 pages
doi:10.1155/2012/246978

Research Article

Simulated Annealing-Based Ant Colony Algorithm for Tugboat Scheduling Optimization

Qi Xu,[1] Jun Mao,[1,2] and Zhihong Jin[1]

[1] *Transportation Management College, Dalian Maritime University, Dalian 116026, China*
[2] *Dalian China Railway International Container Ltd., Dalian 116004, China*

Correspondence should be addressed to Qi Xu, cogi@163.com

Received 5 September 2012; Accepted 23 September 2012

Academic Editor: Rui Mu

As the “first service station” for ships in the whole port logistics system, the tugboat operation system is one of the most important systems in port logistics. This paper formulated the tugboat scheduling problem as a multiprocessor task scheduling problem (MTSP) after analyzing the characteristics of tugboat operation. The model considers factors of multianchorage bases, different operation modes, and three stages of operations (berthing/shifting-berth/unberthing). The objective is to minimize the total operation times for all tugboats in a port. A hybrid simulated annealing-based ant colony algorithm is proposed to solve the addressed problem. By the numerical experiments without the shifting-berth operation, the effectiveness was verified, and the fact that more effective sailing may be possible if tugboats return to the anchorage base timely was pointed out; by the experiments with the shifting-berth operation, one can see that the objective is most sensitive to the proportion of the shifting-berth operation, influenced slightly by the tugboat deployment scheme, and not sensitive to the handling operation times.

1. Introduction

Container terminal is an important part in international logistics and plays a significant role in world trade. Recently, more and more people become to recognize the importance of global logistic business via container terminals. As the throughput of containers in container terminal increases and competition between ports becomes fierce, how to improve the efficiency in container terminal has become an important and immediate challenge for port managers. One of the most important performance measures in container terminals is to schedule all kinds of equipment at an optimum level and to reduce the turnaround time of vessels. Tugboat is one such kind of vital equipments in container terminal.

The performance of the tugboat operation scheduling has a direct influence on time when a ship can start its handling operation and when a ship can leave the port. Scheduling

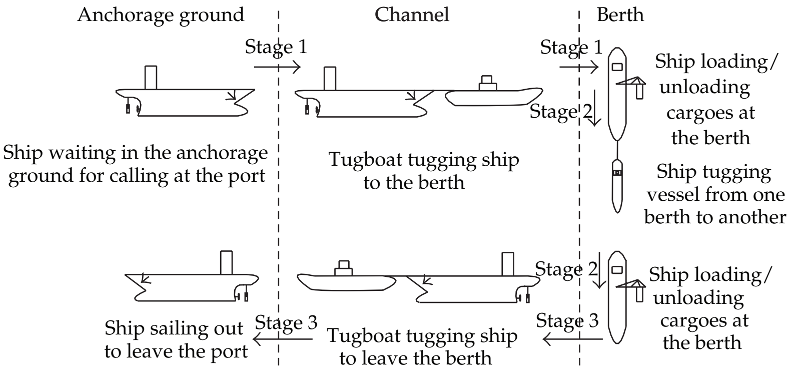

Anchorage ground
Channel
Berth
Stage 1
Stage 1
Ship loading/ unloading cargoes at the berth
Stage 2
Ship waiting in the anchorage ground for calling at the port
Tugboat tugging ship to the berth
Ship tugging vessel from one berth to another
Stage 2
Ship loading/ unloading cargoes at the berth
Ship sailing out to leave the port
Stage 3
Tugboat tugging ship to leave the berth
Stage 3

Figure 1: Illustration of a typical tugboat operation process.

on tugboats with good performance may lower the turnaround of ships in a port. Thus the tugboat scheduling problem is an important one to be solved in the field of the port logistics.

When ships arrive at a port, if their target berths are not available immediately, they cannot enter into the berths directly and have to wait in the anchorage ground. Then they have to be tugged by certain amount of tugboats according to some rules. Moreover, the moving between two berths and the department of vessels also need to be tugged by tugboats. To improve the ship operation efficiency, tugboats should be scheduled at an optimum level.

According to the analysis mentioned above, the three types of service that a tugboat can provide are (a) tugging coming ships to the berth (viz., berthing); (b) tugging ships from one berth to another (namely, shifting-berth); (c) tugging ships leaving the berth (viz., unberthing). Not every ship will experience all the three types of services. That is, the shifting berth operation is not necessary, while the berthing and unberthing operations are necessary for all ships.

A typical tugboat operation process is illustrated in Figure 1. As Figure 1 shows, the duration from the time when a tugboat starts tugging a ship to the finishing time of the berthing operation is treated as stage 1, the duration when a tugboat starts tugging the exact ship leaving the first berth to the finishing time when that ship enter into the second target berth is treated as stage 2, and the duration from the starting time of the unberthing operation to the time when the ship leaves the port is looked upon as stage 3.

Practically, tugboat scheduling managers allocate suitable tugboats to ships according to their length. Each ship can have one or more tugboats serving for it simultaneously by the scheduling rules.

The main idea of the scheduling rules is that big ships should be served by big tugboats (as with the horsepower), and small ships should be served by small tugboats; if more than one tugboat with the same horsepower are available, the allocation among the available tugboats is made by some heuristic rules.

For example, there are six types of tugboats in a port according to the horsepower unit, such as 1200PS, 2600PS, 3200PS, 3400PS, 4000PS, and 5000PS. The scheduling rules for allocating are as follows:

(a) *S1* (less than 100 meter): 1200PS (or bigger)$*$1,

(b) *S2* (100–200 meter): 2600PS (or bigger)$*$2,

(c) *S3* (200–250 meter): 3200PS (or bigger)$*2$,

(d) *S4* (250–300 meter): 3400PS (or bigger)$*2$,

(e) *S5* (greater than 300 meter): 4000PS (or bigger)$*2$.

And the heuristics rules concluded from real-world practice include

(a) TSD rule: choosing the tugboat with the shortest distance from the scheduled ship to serve for it;

(b) FAT rule: choosing the tugboat which is the first available one for the scheduled ship;

(c) UWAT rule: from the perspective of balancing all tugboats' working amount, choosing the tugboat with the minimum working amount up till now to serve for the scheduled ship.

According to the hybrid flow shop theory, the tugboat scheduling can be considered as a multiprocessor task scheduling problem (MTSP) with 3 stages. In the scheduling system, tugboats are taken as movable "machines," and ships have to experience the berthing, shift-berth (if there exists this operation), and unberthing operations operated by tugboats sequentially.

On the other hand, compared with a typical MTSP, the tugboat scheduling problem has its own characteristics. Firstly, the exact same tugboat can provide all the three types of service (berthing, shifting berth, and unberthing), which means that the machine set for all the three stages is the same. This is different from a typical MTSP in which the available machine set in each stage is not the same. Besides, not all ships have to experience the shift-berth operation, which makes the problem different from a typical MTSP with the characteristics that all jobs have to experience all the stages.

Anyway, the tugboat scheduling problem is a kind of unconventional scheduling problem, an NP-hard problem which cannot be solved by exact methods. Some scholars have begun to make research on the topic.

Ying and Lin [1] proposed the ant colony approach to solve the MTSP. Xuan and Tang [2] explored the complexity of the MTSP and designed a Lagrange relaxation algorithm combined with heuristic rules to solve the MTSP. Liu [3] established a mathematical model on the tugboat scheduling problem combined with the MTSP theory and adopted the hybrid evolutionary strategy to solve the model. Liu [4] established an tugboat scheduling model considering the minimum operation distance of the tugboats, and compared the performance of hybrid evolutionary strategy with the particle swarm optimization algorithm for solving the addressed problem. Wang and Meng [5] used a hybrid method that combined ant colony optimization and genetic algorithm to resolve the tugboat allocation problem. Wang et al. [6] formulated a mix-integer model for the tugboat assignment problem combined with the existing scheduling rules and analyzed the effects of the number and service capacity of tugboats on the turnaround time of ships. Liu and Wang [7] considered the tugboat operation scheduling problem as a parallel machine scheduling problem with special process constraint and employed a hybrid algorithm based on the evolutionary strategy to solve the problem. Dong et al. [8] adopted the improved particle swarm optimization combined with dynamic genetic operators to solve the formulated tugboat dispatch model. Liu and Wang [9] used the particle swarm optimization algorithm combined with the local search approach to solve the tugboat scheduling model they proposed.

As we can see from the previous research, scholars have begun to use many approaches to solve the tugboat scheduling problem, including the genetic algorithm, ant colony optimization, and particle swarm optimization. However, most literature only considers the situation of single operation stage and single anchorage base and neglects the influence of the tugboats' and ships' location information on the problem difficulty. That makes the model formulated far from reality. Thus, this paper will make research on tugboat scheduling problem considering multi-anchorage bases, different operation modes, and three operation stages.

The rest of paper is organized as follows. Section 2 formulates a tugboat scheduling model combined with the MTSP theory. Section 3 proposes a simulated annealing-based ant colony algorithm to solve the formulated model, and Section 4 discusses the simulation experiments using ACO in container terminals. Finally, we make conclusions and introduce the future work in Section 5.

2. Model Formulation

2.1. Assumptions

The following assumptions are introduced for the formulation of the problem.

(a) The planning horizon is one day.

(b) Three operation stages (i.e., berthing, shifting-berth, and unberthing) are taken into consideration, but not all ships have to experience the shifting-berth operation. For ship which does not have to experience the second operation, assume there is a virtual shifting-berth operation and the operation time for that is zero.

(c) The ready times for all the tugboats are 0, and all the tugboats are at the anchorage bases at time 0; all the ships to be served have arrived at the anchorage ground at time 0.

(d) There are three types of locations in a port: berths for ships to load/unload cargoes, meeting locations where ships meet tugboats at the entrance of port, and the anchorage bases.

(e) Two operation modes (restricted cross-operation mode RCOM and unrestricted cross operation mode UCOM) may be adopted to schedule the tugboats in a port.

(f) All the ships enjoy the same precedence.

(g) The scheduling rules for allocating tugboats to ships are what we mentioned in Section 1.

(h) The sailing speeds of all tugboats whenever sailing are the same.

(i) The tugboats may return to the anchorage base during the planning horizon according to the scheduling plans.

In assumption (e), the RCOM means that all anchorage bases have their fixed service area in the port, which means that each tugboat in every base can only operate in its corresponding service area, while the UCOM means that all tugboats can operate in the whole area of a port.

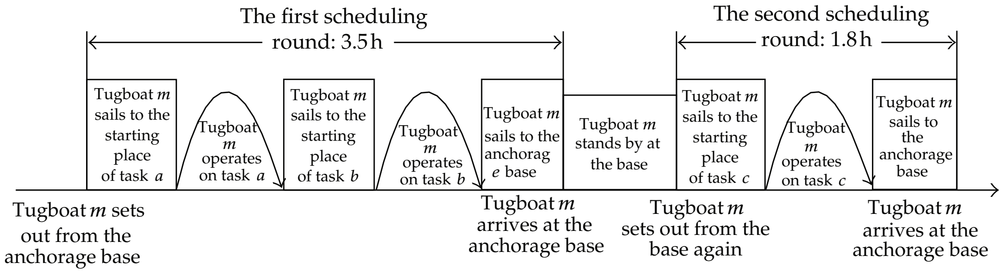

The first scheduling round: 3.5 h
The second scheduling round: 1.8 h
Tugboat m sails to the starting place of task a
Tugboat m operates on task a
Tugboat m sails to the starting place of task b
Tugboat m operates on task b
Tugboat m sails to the anchorage base
Tugboat m stands by at the base
Tugboat m sails to the starting place of task c
Tugboat m operates on task c
Tugboat m sails to the anchorage base
Tugboat m sets out from the anchorage base
Tugboat m arrives at the anchorage base
Tugboat m sets out from the base again
Tugboat m arrives at the anchorage base

Figure 2: Illustration of the tugboat scheduling rounds.

2.2. Definition of the Scheduling Round

Before the tugboat scheduling model is formulated, a concept named scheduling round should be introduced first.

In practice, a scheduling round is used to define the duration from the time when a tugboat leaves for its target place from the anchorage base to the time when it returns to the base after finishing a certain amount of tasks (may be one task, maybe more than one). As Figure 2 illustrates, tugboat m has to operate on three tasks (i.e., a, b, c) in the planning horizon: after finishing the task a, the tugboat sails directly to the starting place of task b and sails back to the anchorage after finishing the task b. That duration can be defined as the first scheduling round of tugboat which lasts for 3.5 hours. On arriving at the anchorage base, the tugboat stands by until it sails to the starting place of task c. After finishing the task c, m sails back to the base again. And that duration from the time when m leaves the base again to the time when it arrives at the base is the second scheduling round which lasts for 1.8 hours. According to the definition, two scheduling rounds occur as to tugboat m in the planning horizon, and the total duration for the two scheduling rounds for tugboat m is 3.5 h + 1.8 h = 5.3 h.

2.3. Notations

(a) Parameters

j, l: Stage index, $j, l \in J = \{1, 2, 3\}$, in which 1–3 represent berthing, shifting-berth, and unberthing operations

i, k: Ship index

cy_i: The descriptive binary parameter that illustrates whether ship i will experience the shifting-berth operation (if $cy_i = 1$, it means that ships i will experience the shifting-berth operation, otherwise will not experience the operation)

m: Tugboat index

M: The set of all the tugboats

ta_m: Style of tugboat m (which may be 1–6, representing 1200PS, 2600PS, 3200PS, 3400PS, 4000PS, and 5000PS, resp.)

N: The set of all ships, $N = \{1, 2, \ldots, n\}$

S_i: Style of ship i

set_i: Set of tugboat style which can provide the related service for ship i

O_{ij}: Operation of ship i at stage j

CM_b: Set of tugboats in the anchorage base b ($b \in B$, B is the set of all the anchorage bases); thus we can get $\bigcup_{b\in B} CM_b = M$

M_{ijb}: Set of tugboats in base b that can serve for operation O_{ij} based on the scheduling rules; thus the set of tugboats in all the bases that can serve for operation O_{ij} can be expressed as $M_{ij} = \bigcup_{b\in B} M_{ijb} = \{m \mid ta_m = \mathrm{set}_i, \forall m \in CM_b\}$

E_{jm}: The set of ships that might be served by tugboat m at stage j

LOS_{ij}: Location where operation O_{ij} starts (if $j = 1$, LOS_{ij} is the meeting place where ship i meets tugboat at the entrance of the port; else if $j = 2$, LOS_{ij} is the first berth where ship i loads/unloads its cargo; else if $j = 3$, LOS_{ij} is the second berth where ship i loads/unloads its cargo, while $\mathrm{LOS}_{i3} = \mathrm{LOS}_{i2}$ if $cy_i = 0$)

LOF_{ij}: Location where operation O_{ij} finishes (if $j = 1$, LOF_{ij} is the first berth where ship$i$ loads/unloads its cargo; else if $j = 2$, LOF_{ij} is the second berth where ship i loads/unloads its cargo, while $\mathrm{LOF}_{i2} = \mathrm{LOF}_{i1}$; else if $j = 3$, LOF_{ij} is the meeting place where ship i meets tugboat at the entrance of the port)

$\mathrm{ST}(a, b)$: Duration for sailing between locations a and b

p_{ij}: Processing time of operation O_{ij}

tb_i: Sailing time of ship i from the waiting place to the berthing place, and $tb_i = \mathrm{ST}(\mathrm{LOS}_{i1}, \mathrm{LOF}_{i2})$

te_i: Berthing time of ship i at berth

toa_i: Duration of ship i for loading and unloading cargoes at the first target berth

tob_i: Duration of ship i for loading and unloading cargoes at the second target berth (if there exists a shifting-berth operation)

tu_i: Unberthing time of ship i at berth

tl_i: Sailing time of ship i from the unberthing place to the place where ship i leaves the port

s^m_{ijkl}: Set-up time between task O_{ij} and O_{kl} by tugboat m

bp_m: The anchorage base where tugboat m belongs

H: A sufficiently large constant.

(b) Decision Variables

$$x_{ijm} = \begin{cases} 1, & \text{if } O_{ij} \text{ is assigned to tugboat } m \\ 0, & \text{otherwise,} \end{cases}$$
$$y_{ijkl}^{m} = \begin{cases} 1, & \text{if } O_{ij} \text{ and } O_{kl} \text{ are assigned to the same tugboat } m \\ 0, & \text{otherwise,} \end{cases}$$
$$u_{ijkl}^{m} = \begin{cases} 1, & \text{if } O_{ij} \text{ precedes } O_{kl} \text{ (not necessarily immediately) on tugboat } m \\ 0, & \text{otherwise,} \end{cases}$$
$$z_{ijkl}^{m} = \begin{cases} 1, & \text{if } O_{ij} \text{ immediately precedes } O_{kl} \text{ on tugboat } m \\ 0, & \text{otherwise,} \end{cases}$$
$$w_{ijm} = \begin{cases} 1, & \text{if tugboat } m \text{ goes back to the anchorage base after completing operation } O_{ij} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

(c) State Variables Decided by Decision Variables

TS_{ij}: The starting time of O_{ij}

TF_{ij}: The finishing time of O_{ij}

BT_{m}: The setting-out time of tugboat m from its anchorage base in the planning horizon

FT_{m}: The returning time of tugboat m after finishing its last task in the planning horizon

sh_{mh}: The duration of the hth scheduling round for tugboat m in the planning horizon

g_m: Number of the scheduling rounds for tugboat m in the planning horizon.

2.4. Model

(a) Objective

In this paper, the objective is to minimize the total operation times of tugboats, which can be equal to the total duration for all the scheduling rounds of all tugboats. Thus we have to derive the calculation method for scheduling rounds.

From the definition of the scheduling round, the relation between the decision variable (w_{ijm}) and the quantity of scheduling rounds in the planning horizon (g_m) can be concluded as follow:

$$g_m = \left|\left\{ w_{ijm} \mid w_{ijm} = 1,\ \forall i \in N,\ \forall j \in J \right\}\right|. \tag{2.2}$$

Equation (2.2) means that the value of g_m equals the times for which tugboat m returns to the anchorage base.

Define the set of tasks right before which tugboat m returns to the base as OS_m, and all the tasks in OS_m are ordered by the operation sequence. By that definition, we come to know the calculation method for duration of each scheduling round of tugboat m as follows:

$$\begin{aligned}
\mathrm{sh}_{m1} &= \mathrm{TF}_{\mathrm{OS}_m\{1\}} + \mathrm{ST}\left(\mathrm{LOF}_{\mathrm{OS}_m\{1\}}, bp\right) - \mathrm{BT}_m \\
\mathrm{sh}_{m2} &= \mathrm{TF}_{\mathrm{OS}_m\{2\}} + \mathrm{ST}\left(\mathrm{LOF}_{\mathrm{OS}_m\{2\}}, bp\right) - \left(\mathrm{TS}_{ij} - \mathrm{ST}\left(bp, \mathrm{LOS}_{ij}\right)\right) \\
&\qquad \left\{(i,j) \mid z^m_{ijkl} = 1, (k,l) = \mathrm{OS}_m\{1\}\right\} \\
&\vdots \\
\mathrm{sh}_{mh} &= \mathrm{TF}_{\mathrm{OS}_m\{h\}} + \mathrm{ST}\left(\mathrm{LOF}_{\mathrm{OS}_m\{h\}}, bp\right) - \left(\mathrm{TS}_{ij} - \mathrm{ST}\left(bp, \mathrm{LOS}_{ij}\right)\right) \\
&\qquad \left\{(i,j) \mid z^m_{ijkl} = 1, (k,l) = \mathrm{OS}_m\{h-1\}\right\} \\
&\vdots \\
\mathrm{sh}_{mg_m} &= \mathrm{TF}_{\mathrm{OS}_m\{g_m\}} + \mathrm{ST}\left(\mathrm{LOF}_{\mathrm{OS}_m\{g_m\}}, bp\right) - \left(\mathrm{TS}_{ij} - \mathrm{ST}\left(bp, \mathrm{LOS}_{ij}\right)\right) \\
&\qquad \left\{(i,j) \mid z^m_{ijkl} = 1, (k,l) = \mathrm{OS}_m\{g_m-1\}\right\}.
\end{aligned} \tag{2.3}$$

Equation (2.3) reveals the duration of each scheduling round equals the finishing time when tugboat completes its last task in the scheduling round plus the sailing time from the location where the last task of tugboat is completed to the tugboat's anchorage base minus the time when tugboat begins its first task in the scheduling round minus the sailing time from the base to the location where the first task starts.

As it has been discussed before, the total operation times of tugboats are equal to the total duration for all the scheduling rounds of all tugboats. Thus the objective function can be expressed as follows:

$$\text{Minimize } F = \sum_{m\in M}\sum_{h\in g_m} \mathrm{sh}_{mh}. \tag{2.4}$$

(b) Constraints

The constraints in the proposed model include the following equations:

$$\mathrm{TS}_{ij} \geq 0, \quad \forall i \in N,\ \forall j \in J, \tag{2.5}$$

$$\begin{gathered}
\mathrm{TS}_{i1} + p_{i1} + toa_i \cdot cy_i \leq \mathrm{TS}_{i2}, \quad \forall i \in N, \\
\mathrm{TS}_{i2} + p_{i2} + tob_i \cdot cy_i + toa_i \cdot \left(1 - cy_i\right) \leq \mathrm{TS}_{i3}, \quad \forall i \in N,
\end{gathered} \tag{2.6}$$

$$\begin{matrix}
\displaystyle\sum_{m\in M} x_{ijm} = 1, & S_i = S1, \\
\displaystyle\sum_{m\in M} x_{ijm} = 2, & \text{otherwise},
\end{matrix} \quad \forall i \in N,\ \forall j \in J, \tag{2.7}$$

$$\begin{gathered}
\text{set}_i = \{1,2,3,4,5,6\} \quad S_i = S1, \\
\text{set}_i = \{2,3,4,5,6\} \quad S_i = S2, \\
\text{set}_i = \{3,4,5,6\} \quad S_i = S3, \\
\text{set}_i = \{4,5,6\} \quad S_i = S4, \\
\text{set}_i = \{5,6\} \quad S_i = S5,
\end{gathered} \tag{2.8}$$

$$\begin{gathered}
y_{ijkl}^{m} \leq 0.5\left(x_{ijm} + x_{klm}\right) \leq y_{ijkl}^{m} + 0.5, \quad \forall i,k \in E_{jm},\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j,l \in J, \\
y_{ijkl}^{m} = y_{klij}^{m}, \quad \forall i,k \in E_{jm},\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j,l \in J,
\end{gathered} \tag{2.9}$$

$$u_{ijkl}^{m} + u_{klij}^{m} = y_{ijkl}^{m}, \quad \forall i,k \in E_{jm},\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j,l \in J, \tag{2.10}$$

$$u_{ijkl}^{m} - z_{klij}^{m} \geq 0, \quad \forall i,k \in E_{jm},\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j,l \in J, \tag{2.11}$$

$$\begin{gathered}
\sum_{k\in E_{lm}} z_{ijkl}^{m} \leq 1, \quad \forall i \in E_{jm},\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j,l \in J, \\
\sum_{k\in E_{lm}} z_{klij}^{m} \leq 1, \quad \forall i \in E_{jm},\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j,l \in J,
\end{gathered} \tag{2.12}$$

$$\begin{gathered}
\text{TS}_{ij} + p_{ij} + s_{ijkl}^{m} \leq \text{TS}_{kl} + H\left(1 - z_{ijkl}^{m}\right), \quad \forall i,k \in N,\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j \in J, \\
\text{TS}_{kl} + p_{kl} + s_{klij}^{m} \leq \text{TS}_{ij} + H\left(1 - z_{klij}^{m}\right), \quad \forall i,k \in N,\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j \in J,
\end{gathered} \tag{2.13}$$

$$\begin{gathered}
p_{ij} = tb_i + te_i, \quad j = 1, \\
p_{ij} = \left(tu_i + \text{ST}(\text{LOS}_{i2}, \text{LOF}_{i2}) + te_i\right) \cdot cy_i, \quad j = 2\ \forall i \in N, \\
p_{ij} = tu_i + tl_i, \quad j = 3,
\end{gathered} \tag{2.14}$$

$$s_{ijkl}^{m} = \text{ST}\left(\text{LOF}_{ij}, \text{LOS}_{kl}\right) \cdot z_{ijkl}^{m} \quad \forall i,k \in N,\ \forall j,l \in J,\ \forall m \in \bigcup_{b\in B} M_{ijb}, \tag{2.15}$$

$$\begin{gathered}
w_{ijm} \cdot H \geq z_{ijkl}^{m} \cdot \left[\left(\text{TS}_{kl} - \text{TF}_{ij}\right) - \left(\text{ST}\left(\text{LOF}_{ij}, bp_m\right) + \text{ST}\left(bp_m, \text{LOS}_{kl}\right)\right)\right], \\
\forall i,k \in E_{jm}, \quad \forall m \in \bigcup_{b\in B} M_{ijb}, \quad \forall j,l \in J, \\
w_{ijm} < \left(z_{ijkl}^{m} \cdot \frac{\text{TS}_{kl} - \text{TF}_{ij}}{2 \times \left(\text{ST}\left(\text{LOF}_{ij}, bp_m\right) + \text{ST}\left(bp_m, \text{LOS}_{kl}\right)\right)} + 0.5\right), \\
\forall i,k \in E_{jm}, \quad \forall m \in \bigcup_{b\in B} M_{ijb}, \quad \forall j,l \in J,
\end{gathered} \tag{2.16}$$

$$x_{ijm}, y_{ijkl}^{m}, u_{ijkl}^{m}, z_{ijkl}^{m}, w_{ijm} = 0 \text{ or } 1, \quad \forall i,k \in N,\ \forall m \in \bigcup_{b\in B} M_{ijb},\ \forall j,l \in J. \tag{2.17}$$

Constraint (2.5) guarantees that each operation begins after time zero. Constraint (2.6) ensures that for every ship, the shifting-berth operation begins only after the berthing and handling operations are completed, and the unberthing operation begins only after the shifting-berth and handling operations are completed. Constraint (2.7) means that if the style of the ship is 1, only one tugboat is needed; otherwise, two tugboats are needed. Constraint (2.8) defines the available set of tugboat style which can serve for ship i according to the scheduling rules. Constraint (2.9) defines $y_{ijkl}^{m} = y_{klij}^{m} = 1$ when $x_{ijm} = x_{klm} = 1$. Constraint (2.10) guarantees that every tugboat can only serve for one operation at any time. Constraint (2.11) is set to make sure that $u_{ijkl}^{m} = 1$ when $z_{ijkl}^{m} = 1$. Constraint (2.12) guarantees that there are at most one predecessor and successor for operation O_{ij} on tugbo at m. Constraint (2.13) simultaneously determines that the starting time of any operation has to be after the time when its immediately preceding task finishes. Constraint (2.14) defines the processing time for each task. Constraint (2.15) defines the set-up time for each operation O_{ij}. Constraint (2.16) simultaneously determines when tugboat m should return to the anchorage base: if the sum of the sailing time from the finishing place of O_{ij} to the base and the sailing time from the base to the starting place of O_{ij}'s successor task on tugboat m (i.e., O_{kl}) is less than the time cost if m directly sails to O_{kl}'s starting place and waits there until the task begins, then tugboat m should return to the base; otherwise, m should sail directly to O_{kl}'s starting place. Constraint (2.17) specifies the binary property of the decision variables.

3. Proposed Hybrid Algorithm (PHA)

3.1. The Basic Idea of the Ant Colony Algorithm

Ant colony metaheuristic is a concurrent algorithm in which a colony of artificial ants cooperates to find optimized solutions of a given problem (see Boveiri [10]). The ant algorithm was first proposed by Dorigo et al. [11] as a multiagent approach to the traveling salesman problem (TSP), and it has been utilized successfully to many difficult discrete optimization n problems such as job shop scheduling, vehicle routing, graph coloring, sequential ordering, and network routing.

The inspiring natural process of ACS is the foraging behavior of ants. A colony of ants can identify the shortest pathway from a food source to their anthill without using visual cues; they communicate through an aromatic substance, called pheromone. While walking, ants secrete pheromone on the ground and follow, in probability, the pheromone previously laid by other ants. Ants are more likely to follow pathways marked by a larger accumulation of pheromone from other ants that have previously walked that route. Since ant searching a food source by shorter pathways will come back to the anthill sooner than ants traveling via longer pathways, the shorter pathways will have a higher traffic density than those of the longer ones. Hence, the pheromone accumulation will build up more rapidly on shorter pathways than on longer ones. Consequently, the fast accumulation of pheromone on the shorter pathways will cause ants to quickly choose the shortest routes. The described foraging behavior of ants can be used to solve scheduling problems by simulation: the objective value (e.g., flow time) corresponds to the quality of the food source (e.g., distance), artificial ants searching for the solution space simulate real ants searching for their environment, and an adaptive memory corresponds to the pheromone trail. In addition, the artificial ants are equipped with a local heuristic function to guide their search through the set of feasible solutions [1].

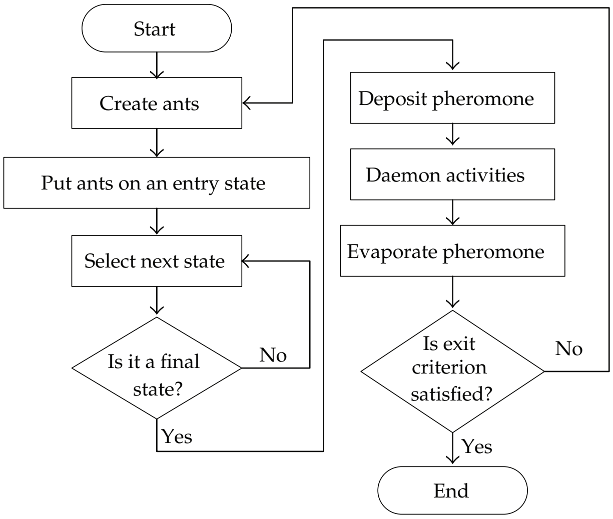


Figure 3: Flowchart of ant colony optimization.

The main procedure of the ant colony algorithm is as follows.

(a) Generate ant (or ants).

(b) Loop for each ant (until complete scheduling of tasks).

 (i) Select the next task with respect to pheromone variables of ready tasks.

(c) Deposit pheromone on visited states.

(d) Daemon activities.

(e) Evaporate pheromone.

The flowchart of ant colony algorithm is illustrated as Figure 3.

However, the solutions were generated by each ant in the basic ant colony algorithm by random, and those solutions may not be the optimal solutions or satisfactory solutions. That makes the updating of the pheromone be done by random too, which may cause a lot of time costs to get the optimal value, and that value may also be the local optima. To avoid that phenomenon, the diversity of the population should be considered.

By that thought, we introduce the simulated annealing into the ant colony algorithm, which can guarantee the quality of the search and avoid the phenomenon of the local optima. Thus, the simulated annealing-based ant colony algorithm is proposed.

3.2. Procedure of the Proposed Algorithm

According to the analysis above, we introduce a simulated annealing-based ant colony algorithm to solve the formulated tugboat scheduling problem. The basic procedure of the algorithm is as Figure 4. In the algorithm, the ACO performs the role of simulation, while the simulated annealing algorithm performs the role of searching for global optimization.

Step 1. Generate the initial tugboat scheduling plans (individuals) which act as representing codes for the simulated annealing algorithm.

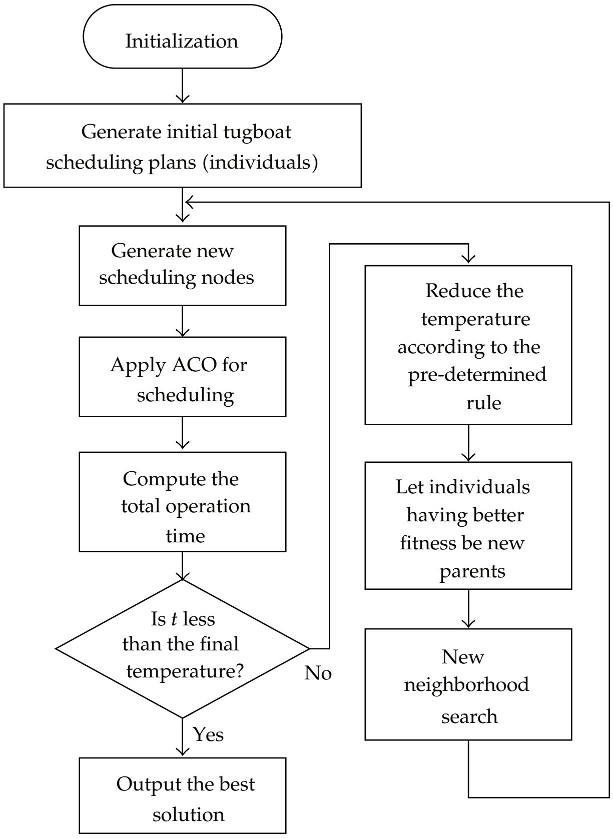

Initialization
Generate initial tugboat scheduling plans (individuals)
Generate new scheduling nodes
Apply ACO for scheduling
Compute the total operation time
Is *t* less than the final temperature?
No
Yes
Output the best solution
Reduce the temperature according to the pre-determined rule
Let individuals having better fitness be new parents
New neighborhood search

Figure 4: Flowchart of the proposed algorithm.

Step 2. Generate new scheduling nodes used to apply for the ant colony algorithm.

Step 3. Apply the ant colony algorithm for the scheduling process.

Step 4. Compute the total operation time for all tugboats in the planning horizon as the key indicator for the system.

Step 5. If the current temperature is less than the final temperature, then go to *Step 9*; else go to *Step 6*.

Step 6. Reduce the temperature according to the predetermined rule.

Step 7. Let the individuals having better fitness be new parents.

Step 8. Based on the new parents, perform a new neighborhood search to get the new individuals.

Step 9. Output the best solution.

3.3. Key Operations of the Algorithm

3.3.1. Ant Colony Optimization (ACO)

In the ant colony algorithm for solving the proposed problem, jobs are defined as ants and resources are defined as nodes. The main procedure of ant colony optimization has been discussed in Section 3.1, and in this section, two key operations of the ACO (i.e., initialization of ants and updating of the pheromone) will be introduced.

(a) Initialization of Ants

According to the algorithm, a certain amount of ants have to be generated. In order to make the schedules by which ants travel satisfy the requirements of the scheduling system, three arrays were set in the algorithm: *tour* which represents tasks not yet operated; *tournext* which represents the tasks to be operated in the next step; *visited* which represents tasks having been operated. All the ants can only choose the tasks for the next operation from the *tournext* array, so that the feasibility of the schedule traveled by ants can be guaranteed. Then, we just need to judge whether all the tasks have been traveled. Then the schedule generated by ants in the *visited* is the schedule we want.

The selection of nodes during the algorithm is referenced by the roulette wheel. Thus the state transit rule can be concluded as.

$$p_{ij} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[1/T_{ij}]^{\beta}}{\sum_{s\in \text{allowed}}[\tau_{ij}(t)]^{\alpha}[1/T_{ij}]^{\beta}} \\ 0. \end{cases} \tag{3.1}$$

In (3.1), T_{ij} means the processing time of ship i by tugboat j, *allowed* = *tournext*.

(b) Updating of the Pheromone

After all ants of a generation have traveled all the tasks, compute the total operation times *o* the tugboats and update the pheromone according to those values. In our research, we choose the five ants with the minimal operation times to update the pheromone, and the updating rules are as follows:

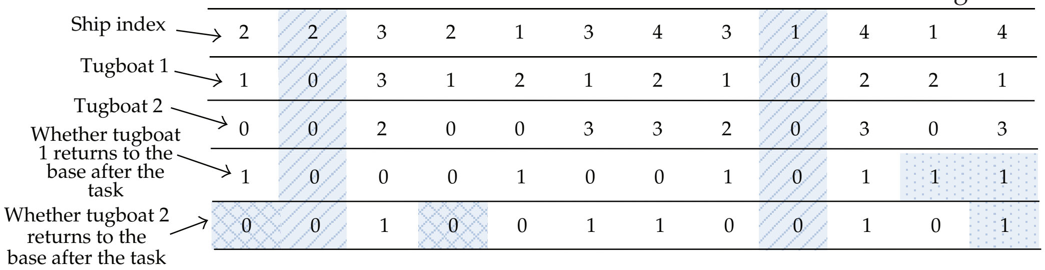
$$\tau_{ij} = (1-\rho)\tau_{ij} + V_{\tau_{ij}}, \tag{3.2}$$

$$V_{\tau_{ij}} = \frac{Q}{T_{\min}}. \tag{3.3}$$

In (3.2), ρ means the evaporation coefficient, $T_{\min}$ in (3.3) means the minimal operation times of all ants in the current generation, and Q is the quantity of pheromone in the unit path.

3.3.2. Simulated Annealing (SA)

The key operations in the simulated annealing include individual coding, initial individuals' generation, and the neighborhood search scheme.



Ship index	2	2	3	2	1	3	4	3	1	4	1	4
Tugboat 1	1	0	3	1	2	1	2	1	0	2	2	1
Tugboat 2	0	0	2	0	0	3	3	2	0	3	0	3
Whether tugboat 1 returns to the base after the task	1	0	0	0	1	0	0	1	0	1	1	1
Whether tugboat 2 returns to the base after the task	0	0	1	0	0	1	1	0	0	1	0	1

Figure 5: Illustration of coding for an individual.

(a) Individual Coding

In this paper, the real integer method is adopted to code for an individual. As every ship may experience at most 3 stages of operation, we set the number of columns as three times of the number of ships. Assume that there are 4 ships to be served (ship 1, 2 do not have to shift a berth, while ship 3, 4 will experience a shifting-berth operation) and 3 available tugboats, then the coding expression of the individuals should be a 5 × (3 × 6) matrix, which can be illustrated as Figure 5.

The first row of the coding representation means the service order for ships, and the next two rows are the indexes of tugboats serving for ships in the first row. Note that each index appears three times in the first row: if it is the first time an index appears, it means that the ship is berthing; for the second time it appears, it may be a virtual or real shifting-berth operation; otherwise, the unberthing operation. The fourth and fifth rows are descriptive parts which tell us whether tugboat 1 and 2 return to the base after finishing the task.

As ship 1 and 2 do not have to shift a berth, the virtual shifting-berth operations are proposed to keep the total operations three times of the number of ships. That can be illustrated as the shadow parts with diagonal lines in Figure 5. Besides, if the ship style is 1, then an index of tugboat is generated from the available tugboat set to fill in the corresponding second row, and the third row is zero (as shown in the shadow parts with grids); otherwise, two indexes of tugboats are generated to fill in the two rows. Thirdly, as all tugboats have to return to the base after finishing their last tasks, the corresponding symbols in the fourth or fifth rows should be 1 (as the shadow parts with dots).

According to that individual coding, the service order for ships in Figure 5 is as follows: ship 2 (berthing)—ship 2 (virtual shifting-berth)—ship 3 (berthing)—ship 2 (unberthing)—ship 1 (berthing)—ship 3 (shifting-berth)—ship 4 (berthing)—ship 3 (unberthing)—ship 1 (virtual shifting-berth)—ship 4 (shifting-berth)—ship 1 (unberthing)—ship 4 (unberthing). The tugboat providing the berthing service for ship 2 is tugboat 1, and after finishing the berthing service for ship 2, tugboat 1 returns to the anchorage base, and so on.

(b) Initial Individuals' Generation

The procedure for generating the initial schedule can be described as Figure 6.

As we can see from Figure 6, the procedure for the initial individuals' generation mainly include three parts: randomly generating the service order for ships; allocating the tugboat serving for ships; deciding whether tugboats should return to the base after completing the operation.

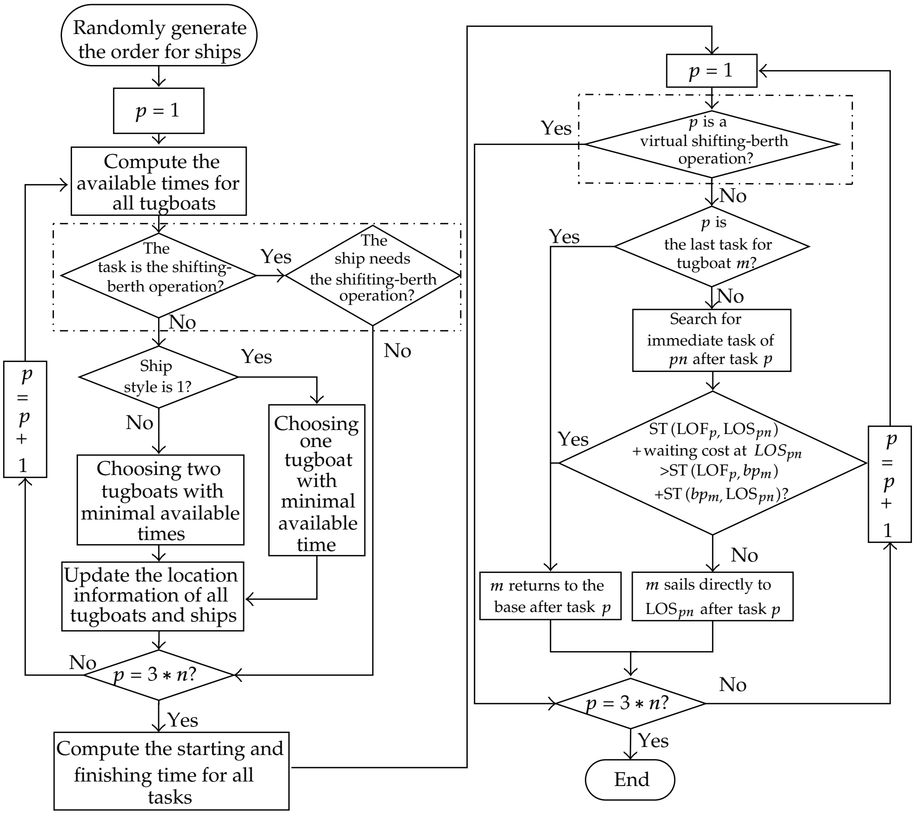

Randomly generate the order for ships
$p = 1$
Compute the available times for all tugboats
The task is the shifting-berth operation?
Yes
The ship needs the shifiting-berth operation?
No
No
Ship style is 1?
Yes
No
Choosing two tugboats with minimal available times
Choosing one tugboat with minimal available time
Update the location information of all tugboats and ships
$p = p + 1$
No
$p = 3 * n$?
Yes
Compute the starting and finishing time for all tasks
$p = 1$
Yes
p is a virtual shifting-berth operation?
No
Yes
p is the last task for tugboat m?
No
Search for immediate task of pn after task p
Yes
ST (LOF$_p$, LOS$_{pn}$) +waiting cost at LOS_{pn} >ST (LOF$_p$, bp_m) +ST (bp_m, LOS$_{pn}$)?
No
$p = p + 1$
m returns to the base after task p
m sails directly to LOS$_{pn}$ after task p
No
$p = 3 * n$?
Yes
End

Figure 6: The generation procedure of the initial individuals.

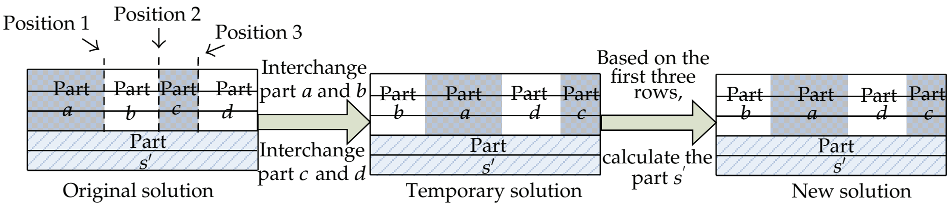
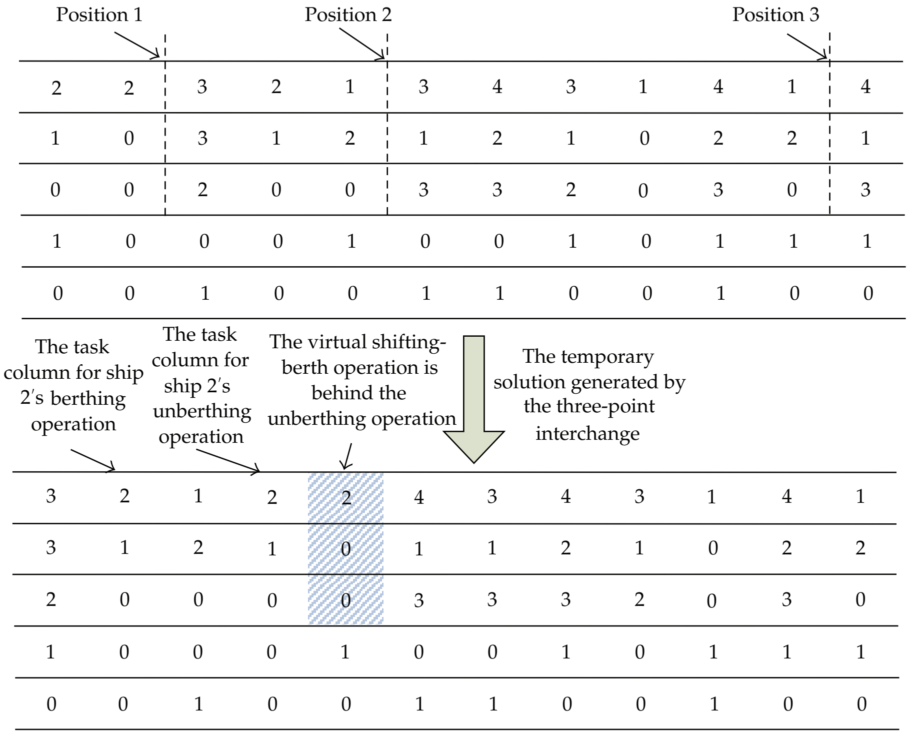


Figure 7: The neighborhood search scheme.

(c) Neighborhood Search Scheme

The procedure for the neighborhood search scheme can be concluded as Figure 7.

Given a solution p, a neighbor of p can be obtained by using the three-point interchanging scheme proposed in this section. The main idea is as follows: randomly generate three positions in the original solution, so that the original solution is divided into five parts; let a, b, c, d, s be the four partial solutions of p; a temporary solution is obtained by interchanging a and b, c and d; based on the three rows of the temporary solution, calculate part s' according to the rules expressed by (2.16).


Position 1
Position 2
Position 3
The task column for ship 2′s berthing operation
The task column for ship 2′s unberthing operation
The virtual shifting-berth operation is behind the unberthing operation
The temporary solution generated by the three-point interchange

2	2	3	2	1	3	4	3	1	4	1	4
1	0	3	1	2	1	2	1	0	2	2	1
0	0	2	0	0	3	3	2	0	3	0	3
1	0	0	0	1	0	0	1	0	1	1	1
0	0	1	0	0	1	1	0	0	1	0	0

3	2	1	2	2	4	3	4	3	1	4	1
3	1	2	1	0	1	1	2	1	0	2	2
2	0	0	0	0	3	3	3	2	0	3	0
1	0	0	0	1	0	0	1	0	1	1	1
0	0	1	0	0	1	1	0	0	1	0	0

Figure 8: The infeasible solution generated by the three-point interchange.

However, during the neighborhood search process, the temporary solution may be an infeasible solution. For example, the virtual shifting-berth operation (the shadow parts in Figure 8) is after the unberthing operation, which is infeasible.

Thus it is necessary to modify the temporary solution. Steps for modifying the temporary solution are as follows.

Step 1. Initialize $p = 1$.

Step 2. Judge if the second and third rows of the pth column are both zero.

(a) If both the values are zero, which means that the task in the pth column is a virtual shifting-berth operation,

(i) search for two columns: one for the berthing operation for ship served in the pth column; one for the unberthing operation for the same ship. Define the places of the two columns as $p1$ and $p2$,

(ii) if p is less than $p1$, interchange values of the first three rows in the two columns, then go to Step 3,

(iii) if p is larger than $p2$, interchange values of the first three rows in the two columns, then go to Step 3.

(b) If the two values are not both zero, then go to *Step 3*.

Table 1: Sailing times between each location.

	P1	P2	P3	P4	P5	P6	P7	P8	M1	M2	B1	B2
P1	0	18	14	20	32	34	35	30	19	29	15	32
P2	18	0	23	15	31	33	27	35	21	33	12	35
P3	14	23	0	12	39	34	30	32	15	38	16	31
P4	20	15	12	0	35	38	31	39	12	31	18	34
P5	32	31	39	35	0	18	12	19	31	12	29	11
P6	34	33	34	38	18	0	13	15	34	11	36	15
P7	35	27	30	31	12	13	0	12	29	18	25	19
P8	30	35	32	39	19	15	12	0	33	15	39	12
M1	19	21	15	12	31	34	29	33	0	30	15	25
M2	29	33	38	31	12	11	18	15	30	0	28	16
B1	15	12	16	18	29	36	25	39	15	28	0	26
B2	32	35	31	34	11	15	19	12	25	16	26	0

Step 3. Judge if p is equal to $3 * n$:

(a) if p is equal to $3 * n$, then the modification is completed;

(b) else, set $p = p + 1$, and go to *Step 2*.

After being modified according to the steps introduced above, the temporary solution can be changed to a new solution by deciding whether tugboats should return to the base according to (2.16).

4. Computational Experiments

4.1. Experimental Data

To implement a comparison of the findings from the proposed algorithm, some experimental data were randomly generated, details of which are as follows.

(a) Location data: the sailing times between each location (P1–P8, M1-M2, B1-B2) are as Table 1. Therein, P1–P8 are locations of 8 berths; M1 is the location where ships whose target berths P1–P4 meet tugboats at the entrance of port; M2 is the location where ships whose target berths P5-P5 meet tugboats at the entrance of port; B1 and B2 are two anchorage bases of tugboats whose service area are P1–P4 and P5–P8, respectively.

(b) Ship data: styles of ships are generated to S1, S2, S3, S4, and S5 which take up about 10%, 20%, 40%, 20%, and 10% of the total ships, berthing/unberthing times, loading and unloading times of ships are normally distributed in $N(35,25)$, $N(300,1600)$, and the berthing locations of ships are uniformly distributed to P1–P8.

(c) Tugboat data: quantities of the six kinds of tugboats in the two anchorage bases are all one.

Table 2: Results of the PHA versus existing scheduling rules.

Number of ships	RCOM				UCOM			
	PHA	TSD	FAT	UWAT	PHA	TSD	FAT	UWAT
10	3743	4222	4181	4280	3743	4200	4156	4245
15	4983	5551	5504	5609	4951	5485	5456	5575
20	6800	7436	7357	7483	6705	7353	7242	7451
25	8481	9132	9030	9220	8405	8900	8858	9100
30	10012	11185	10993	11530	9988	11031	10920	11235

Table 3: Results from the proposed algorithm under two different operation modes.

Number of ships	Total operation times of all tugboats			Average operation times of all tugboats		
	RCOM	UCOM	GAP[1]	RCOM	UCOM	GAP[2]
10	3743	3743	0	312	312	0
15	4983	4951	32	415	413	3
20	6800	6705	95	567	559	8
25	8481	8405	76	707	700	6
30	10012	9988	24	834	832	2

[1] Operation times of all tugboats under the RCOM—values under the UCOM; [2] average operation times of all tugboats under the RCOM—values under the UCOM.

4.2. Experiments without the Shifting-Berth Operation

Suppose the basic data are as shown in Section 4.1, no shifting-berth operation exists and all tugboats do not return to the anchorage base during the planning horizon, use the proposed hybrid algorithm to solve the established tugboat scheduling problem, and then we can get the performance comparisons of the hybrid algorithm with three existing scheduling rules with different number of ships, which are shown in Table 2.

As we can see from Table 2, the PHA's solved results are all far less than those from the three existing scheduling rules. All those can fully describe the efficiency of the PHA. Besides, the performance of the three scheduling rules reveals the same rules: FAT is superior to TSD, and TSD is better than UWAT. That is because the FAT rule considers both the TSD and UWAT rules, while the UWAT rule only considers the uniform scheduling of every tugboat but might cause the postponement of the waiting time of ships for tugboats.

Besides, we can see from Table 3 that the difference between the two operation modes increases from zero and then decreases to near zero. The reason for that phenomenon can be concluded as follows: when the number of ships is small and the available tugboats are abundant, the optimal solution is the scheduling scheme under the RCOM (as the solutions under the UCOM include those under the RCOM, so the optimal solution and optimal value of the modes are the same), which means there is no need to transfer tugboats from another anchorage base; as the number of the ships increases and the tugboat resource becomes scarce, the cost for ships to wait for unoccupied tugboats in another anchorage base is less than that of waiting for busy tugboats in its own anchorage base, so the UCOM is better than RCOM; while the number of ships is great and all tugboats in both anchorage bases are busy, there is no point of transferring tugboats from another anchorage base, which means the RCOM is better.

After the basic analysis above, we compare the operation times on whether tugboats return to the anchorage base during the planning horizon, the results of which can be shown as Table 4.

Table 4: Comparisons between the total operation times on whether tugboats return to the base.

Number of ships	RCOM			UCOM		
	Results if tugboats do not return to the base $(f1)$	Results if tugboats return to the base $(f2)$	GAP*	Results if tugboats do not return to the base $(f1)$	Results if tugboats return to the base $(f2)$	GAP*
10	3743	2675	39.93%	3743	2675	39.93%
15	4983	4005	24.42%	4951	3952	25.28%
20	6800	5285	28.67%	6705	5206	28.79%
25	8481	6575	28.99%	8405	6503	29.25%
30	10012	8007	25.04%	9988	7951	25.62%

*Percentage that $f1$ is larger than that of $f2$, which can be calculated by $(f1 - f2)/f2 \times 100\%$.

Table 5: Results with different proportion of the shifting berth operation.

Number of ships	Results with different proportion of the shifting-berth operation								
	0%[1]	5%[2]	GAP1[a]	10%[3]	GAP2[b]	15%[4]	GAP3[c]	20%[5]	GAP4[d]
10	2675	2845	6.36%	3076	13.04%	3224	20.52%	3415	27.66%
15	3952	4205	6.40%	4518	12.53%	4703	19.00%	4961	25.53%
20	5206	5485	5.36%	5824	10.61%	6121	17.58%	6452	23.93%
25	6503	6857	5.44%	7331	11.29%	7715	18.64%	8035	23.56%
30	7951	8381	5.41%	9122	12.84%	9423	18.51%	9731	22.39%
Average	/	/	5.79%	/	12.06%	/	18.85%	/	24.61%

[a] (value of 2 − value of 1)/value of 1 × 100%; [b] (value of 3 − value of 1)/value of 1 × 100%; [c] (value of 4 − value of 1)/value of 1 × 100%; [d] (value of 5 − value of 1)/value of 1 × 100%.

Based on Table 4, we can see that the operation times if tugboats do not return to the base are 30% larger than those if tugboats return to the base during the planning horizon. That means if tugboats do not return to the base during the horizon, there exists at least 30% of the total sailing routes which are ineffective sailing routes, which is infeasible and does not coincide with the modern concept of green transportation.

4.3. Experiments with the Shifting-Berth Operation

In this section, sensitivity analysis of the three elements to the objective is to be made, and all the experiments done are under the UCOM mode and based on the assumption that tugboats can return to the anchorage base during the planning horizon.

(a) Sensitivity Analysis of the Proportion of the Shifting-Berth Operation

Assume that there are 0%, 5%, 10%, 15%, 20% of the total ships which have to experience the shifting-berth operation, the minimal total operation times of all tugboats when the number of ships is 10, 15, 20, 25, 30 are summarized in Table 5.

As we can see from Table 5, the GAPs (a, b, c, d) are all larger than the proportion of the shifting-berth operation. That is because a single shifting-berth operation contains an unberthing operation, a shift between the berths, and a berthing operation, thus needs more tugboats' resource than normal berthing and unberthing operations. So it is necessary

Table 6: Results with different distribution characteristics of the handling operation times.

Number of ships	Results with different distribution characteristics of the handling operation times				
	$N(300, 1600)^1$	$N(350, 2500)^2$	GAP1[a]	$N(400, 3600)^3$	GAP2[b]
10	2845	2815	−1.05%	2862	0.60%
15	4205	4240	0.83%	4200	−0.12%
20	5485	5522	0.67%	5488	0.05%
25	6857	6870	0.19%	6840	−0.25%
30	8381	8387	0.07%	8428	0.56%
Average	/	/	0.14%	/	0.17%

[a] (value of 2 − value of 1)/value of 1 × 100%; [b] (value of 3 − value of 1)/value of 1 × 100%.

to reduce the number of shifting-berth operation in practice, so that the full utilization of limited tugboat resources.

(b) Sensitivity Analysis of the Distribution Characteristics of the Handling Operation Times

Assume that the distribution characteristics of handling operation times of ships at berth are $N(300, 1600)$, $N(350,2500)$, and $N(400, 3600)$, the proportion of the shifting-berth operation is 5%. The results by the PHA are concluded in Table 6.

As we can see from Table 6, there is no obvious trend about the total operation times according to the changing of the handling times of ships at berth. That is to say, the objective function is not sensitive to the change of the handling times. The reason for that phenomenon can be concluded as follows.

Compared with the operation times of tugboats, the handling times are much larger. After completing a certain task, a tugboat can return to the base to have a rest and then sail to its next target location. With the increase of the handling operation times, the wait times in the base may also increase, which are not parts of the total operation times of tugboats. Thus, the objective does not reveal obvious reaction to the change of the handling times.

(c) Sensitivity Analysis of the Tugboat Deployment Scheme

We then assume different deployment schemes of the available tugboats in the port (i.e., Scheme 1: the number of all types are 1; Scheme 2: the number of type 6 are 2, others are 1; Scheme 3: the number of type 5 and 6 are 2, others are 1). The results solved by the PHA are summarized in Table 7. Therein, the proportion of the shifting-berth operation is still 5%.

As Table 7 shows, the total operation times of all tugboats reveal a mild trend of decrease as the number of tugboats deployed increases. That is to say, the total operation times of tugboats can only be slightly reduced by simply increasing the number of tugboats deployed, and the cost of increasing tugboats may well be larger than the time cost saved by that. Under that circumstance, adding extra tugboats is not advised.

By the analysis, we can say that the objective is most sensitive to the proportion of the shifting-berth operation, influenced slightly by the tugboat deployment scheme, and not sensitive to the handling operation times.

Table 7: Results with different tugboat deployment schemes.

Number of ships	Results with different tugboat deployment schemes				
	Scheme 1[1]	Scheme 2[2]	GAP1[a]	Scheme 3[3]	GAP2[b]
10	2845	2833	–0.42%	2808	–1.30%
15	4205	4186	–0.45%	4159	–1.09%
20	5485	5421	–1.17%	5394	–1.66%
25	6857	6807	–0.73%	6785	–1.05%
30	8381	8325	–0.67%	8299	–0.98%
Average	/	/	–0.69%	/	–1.22%

[a] (value of 2 – value of 1)/value of 1 × 100%; [b] (value of 3 – value of 1)/value of 1 × 100%.

5. Concluding Remarks

This paper formulated the tugboat scheduling problem as a multiprocessor task scheduling problem (MTSP). The model considers factors of multi-anchorage bases, different operation modes, and three stages of operations (berthing/shifting-berth/unberthing). A hybrid simulated annealing-based ant colony algorithm is proposed to solve the addressed problem. By the numerical experiments without the shifting-berth operation, the effectiveness were verified, and the fact that more effective sailing may be possible if tugboats return to the anchorage base timely was pointed out; by the experiments with the shifting-berth operation, the paper proved that the objective is most sensitive to the proportion of the shifting-berth operation, influenced slightly by the tugboat deployment scheme, and not sensitive to the handling operation times.

Future work about the topic should be to extend the problem from the static situation to a dynamic one, although it may be much more difficult but more meaningful.

References

[1] K. C. Ying and S. W. Lin, “Multiprocessor task scheduling in multistage hybrid flow-shops: an ant colony system approach,” *International Journal of Production Research*, vol. 44, no. 16, pp. 3161–3177, 2006.

[2] H. Xuan and L. X. Tang, “Dynamic hybrid flowshop scheduling problem with multiprocessor tasks,” *Computer Integrated Manufacturing Systems*, vol. 13, no. 11, pp. 2254–2288, 2007.

[3] Z. Liu, “Hybrid evolutionary strategy optimization for port tugboat operation scheduling,” in *Proceedings of the 3rd International Symposium on Intelligent Information Technology Application (IITA ’09)*, pp. 511–515, November 2009.

[4] Z. Liu, “Port tugboat operation scheduling optimization considering the minimum operation distance,” *Journal of Southwest Jiaotong University*, vol. 46, no. 5, pp. 875–881, 2011.

[5] S. Wang and B. Meng, “Resource allocation and scheduling problem based on genetic algorithm and ant colony optimization,” in *Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD ’07)*, vol. 4426 of *Lecture Notes in Computer Science*, pp. 879–886, Nanjing, China, 2007.

[6] S. Wang, I. Kaku, G. Chen, and M. Zhu, “Research on the modeling of Tugboat Assignment Problem in container terminal,” *Advanced Materials Research*, vol. 433–440, pp. 1957–1961, 2012.

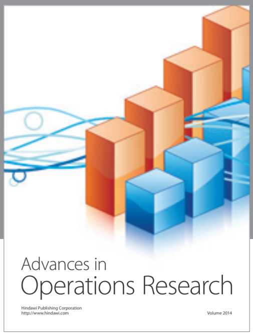
[7] Z. X. Liu and S. M. Wang, “Research on bi-objectives parallel machines scheduling problem with special process constraint,” *Computer Integrated Manufacturing Systems*, vol. 11, no. 11, pp. 1616–1620, 2005.

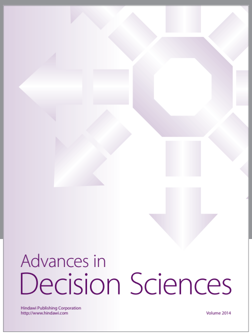
[8] L. C. Dong, Z. Q. Xu, and W. J. Mi, “The dynamic tugboat schedule based on particle swarm algorithm combined with genetic operators,” *Mathematics in Practice and Theory*, vol. 42, no. 6, pp. 122–133, 2012.

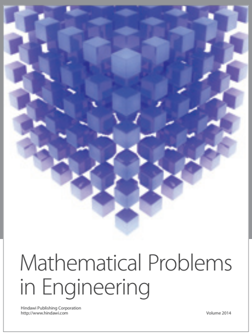
[9] Z. X. Liu and S. M. Wang, “Research on parallel machines scheduling problem based on particle swarm optimization algorithm,” *Computer Integrated Manufacturing Systems*, vol. 12, no. 2, pp. 183–296, 2006.

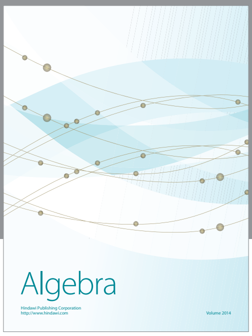
[10] H. R. Boveiri, “ACO-MTS: a new approach for multiprocessor task scheduling based on ant colony optimization,” in *Proceedings of the International Conference on Intelligent and Advanced Systems (ICIAS '10)*, pp. 175–179, June 2010.

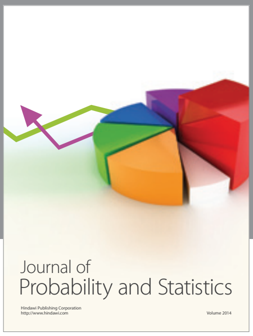
[11] M. Dorigo, G. DiCaro, and L. Gambardella, “Ant algorithm for discrete optimization,” *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.


Advances in
Operations Research

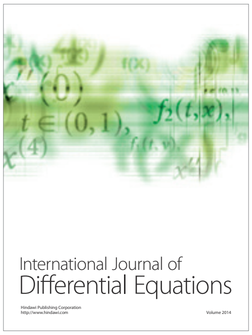

Advances in
Decision Sciences

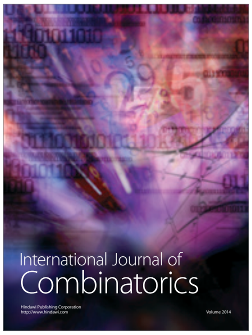

Mathematical Problems
in Engineering

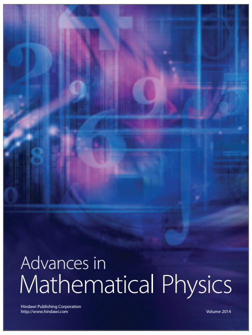

Algebra

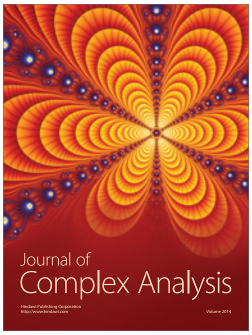

Journal of
Probability and Statistics

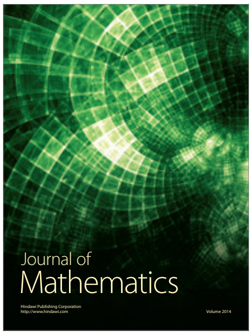
The Scientific
World Journal

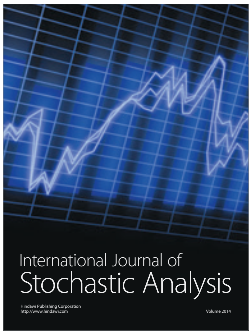

International Journal of
Differential Equations

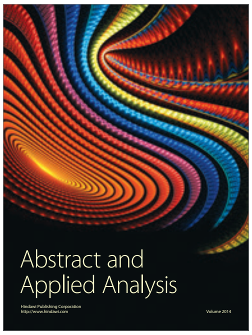

International Journal of
Combinatorics

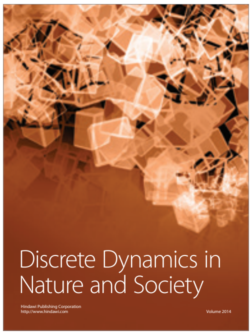

Advances in
Mathematical Physics

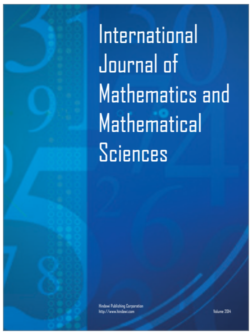

Journal of
Complex Analysis


Journal of
Mathematics


International Journal of
Stochastic Analysis

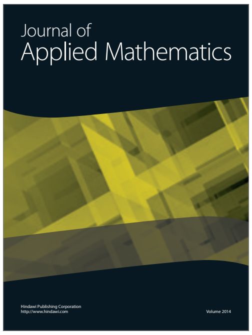

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


Journal of
Applied Mathematics

Journal of
Optimization