Hindawi
Scientific Programming
Volume 2018, Article ID 3450726, 9 pages
https://doi.org/10.1155/2018/3450726

Research Article

Stowage Planning in Multiple Ports with Shifting Fee Minimization

E. Zhang,[1] Qihuang Mei ,[2] Ming Liu ,[3] and Feifeng Zheng[2]

[1] *School of Information Management & Engineering, Shanghai University of Finance and Economics, Shanghai, China*
[2] *Glorious Sun School of Business & Management, Donghua University, Shanghai, China*
[3] *School of Economics & Management, Tongji University, Shanghai, China*

Correspondence should be addressed to Ming Liu; mingliu@tongji.edu.cn

Received 21 August 2017; Revised 20 December 2017; Accepted 10 January 2018; Published 1 February 2018

Academic Editor: Emiliano Tramontana

This paper studies the problem of stowage planning within a vessel bay in a multiple port transportation route, aiming at minimizing the total container shifting fee. Since the access to containers is in the top-to-bottom order for each stack, reshuffle operations occur when a target container to be unloaded at its destination port is not stowed on the top of a stack at the time. Each container shift via a quay crane induces one unit of shifting fee that depends on the charge policy of the local container port. Previous studies assume that each container shift consumes a uniform cost in all ports and thus focus on minimizing the total number of shifts or the turnaround time of the vessel. Motivated by the observation that different ports are of nonuniform fee for each container shift, we propose a mixed integer programming (MIP) model for the problem to produce an optimal stowage planning with minimum total shifting fee in this work. Moreover, as the considered problem is NP-hard due to the NP-hardness of its counterpart with uniform unit shifting fee, we propose an improved genetic algorithm to solve the problem. The efficiency of the proposed algorithm is demonstrated via numerical experiments.

1. Introduction

In the world seaborne trade, most goods are transported by containerships. Containership maritime transports have occupied the vast majority of the maritime transport industry, among which container trade is now becoming the fastest-growing freight segment (see [1]).

An efficient stowage planning is one of the most important factors in saving the transportation cost of shipping companies. One vessel may visit several container ports during a voyage, and containers are loaded onto the vessel from some upstream ports and later unloaded from the vessel at downstream ports. If a target container to be unloaded from the vessel is not stowed on the top of a stack, it incurs reshuffle operations as the access to containers follows the top-to-bottom order for any stack. All the involved containers piled above the target one have to be removed temporarily and later reloaded back to the stack after the target container has been unloaded. The unloading/reloading operations of the involved nontarget containers cause extra shifting fees as well as time consumption. An efficient stowage solution to the vessel may greatly reduce such shifting fees in a transportation route.

Most previous studies aim to minimize the total number of shifts for nontarget containers at each container port and therefore to minimize the turnaround time of a vessel. By field investigation, however, we observe that the shifting fee is also an important consideration, and a key point is that the unit shifting fee varies vastly at different container ports. For example, the unit shifting fee for one container move at Ningbo-Zhoushan port is 49.5 RMB, while it is 100 RMB at Guangzhou port. The observation motivates us to investigate the stowage planning problem such that the total shifting fee other than the total number of shifts for a vessel is minimized in a multiple port transportation route. As the containers in each bay of the vessel are individually handled at each port, we consider the stowage planning within a bay on the vessel. We claim that the stowage planning problem under consideration is NP-hard since its counterpart with uniform unit shifting fee, that is, the problem to minimize the number of shifts, is already NP-hard [2]. We have not found any related work on total shifting fee minimization for the case with nonuniform unit shifting fee. Aiming at this problem, we establish an MIP model and design an efficient algorithm to produce good stowage planning solutions.

The rest of this work is organized as follows. Section 2 gives a brief literature review of related works, and Section 3 describes the considered problem formally. In Section 4, we present the MIP mathematical formulation, and in Section 5 we propose a genetic algorithm for the problem. Numerical results are given in Section 6, and finally we conclude the paper in Section 7.

2. Literature Review

In recent decades, the problem of containership stowage planning has been extensively investigated [3–6].

2.1. Stowage Planning in Single Port. Delgado et al. [7] study the stowage planning problem and propose an approach that generates a near optimal plan for a large container vessel within a few minutes. The approach solves the problem in two steps. In the first step it decomposes the vessel into master bays and assigns containers to the master bays. In the second step, the containers in each master bay are further assigned to specific slots of the bay.

Monaco et al. [8] are involved in the problem of determining the optimal positions of containers to be stowed in a vessel. They assume that the vessel berthing along the quay consists of a number of slots and propose a binary integer program and a two-step heuristic algorithm to obtain efficient solutions to the stowage planning problem.

Ambrosino et al. [9] study the Master Bay Plan Problem (MBPP) and propose a tabu search metaheuristic approach to look for the global ship stability of the overall stowage plan. The proposed tabu search algorithm can improve planning management in the visualization of stowage plans in a software support system. It also performs well on some real-life test cases related to a terminal located at the port of Genoa, Italy.

Liu et al. [10] investigate the quay crane double-cycling problem with internal-reshuffling operations and present a polynomial-time heuristic algorithm to reduce the number of container moves in a bay of a vessel. By comparing their algorithm with the state-of-the-art heuristic, they demonstrate that their model can be solved more efficiently than the one proposed in Meisel and Wichmann [11].

2.2. Stowage Planning in Multiple Port. Avriel et al. [2] deal with a stowage plan for container ships to minimize the number of shifts and first present a binary linear programming formulation to find optimal solutions for stowage planning. Due to many binary variables and constraints, they develop the so-called Suspensory Heuristic Procedure.

Araújo et al. [12] consider a three-dimensional container ship stowage planning problem. They develop a mathematical model with the objective of minimizing the number of container moves and ship instability. A hybrid method is proposed to solve the model and obtain a good approximation to the Pareto front. Computational results reveal that the proposed method provides better solutions than the monoobjective simulated annealing algorithm.

Ding and Chou [13] consider the stowing planning problem where a vessel visits a series of ports sequentially. They develop a heuristic algorithm for generating stowage plans with a reasonable number of container shifts and show that the algorithm performs better than the Suspensory Heuristic Procedure proposed in Avriel et al. [2].

Ambrosino et al. [4] extend the MBPP to the Multiport Master Bay Plan Problem (MP-MBPP), and propose a heuristic algorithm based on an exact MIP model to minimize the total berthing time of the vessel. The proposed efficient heuristic algorithm can find good solutions to the whole trip planning.

The above literature analyzes both single-port and multiport container loading/unloading scenarios. For more results on the stowing planning problem, please refer to Wang [14] and Ramos et al. [15]. All these previous works concern the minimum reshuffles or turnaround times in the route; that is, they all assume that each container shift is of identical cost in all the ports. Some authors also build linear programming models for their respective problems [2, 12]. However, as we have previously mentioned, different ports are usually of nonuniform operational costs and unit shifting fee of a container in practice. Thus we establish a mixed integer programming model for the case with nonuniform unit shifting fee together with load balance requirement in this work. To the best of our knowledge, there are no relevant studies in literature that consider the case with nonuniform-shifting fee and aiming to minimize the total cost of shifting containers in a route.

3. Problem Description

We consider the stowage planning problem on a container vessel in multiple ports and focus on the container shifting operations within one of the bays on the vessel. The bay consists of a number C of container columns or stacks where there may be at most R containers stowed in each vertical stack. Each container occupies one layer of the stack or equally a slot of the bay. We assume that all the containers are TEU (Twenty-foot Equivalent Unit) ones. Figure 1 is a two-dimensional coordinate illustration where there is a side view and a bird's eye view of the containers on a vessel. A more detailed description of container bays on the vessel is referred to by Ambrosino et al. [3]. The slot location of each container in the bay is denoted by the combination of column and layer indices, that is, slot (c, r) where $1 \le c \le C$, $1 \le r \le R$.

The vessel departs from the first port $t = 1$ and sequentially visits ports $t = 2, \dots, T-1, T$. Initially, there are no containers in the bay before loading at the first departure port. Notice that there are no container shifts at $t = 1$ port since there are only loading containers. Similarly, at the final port T, all the remaining containers on the vessel are to be unloaded from the vessel, and there exist no container shifts either.

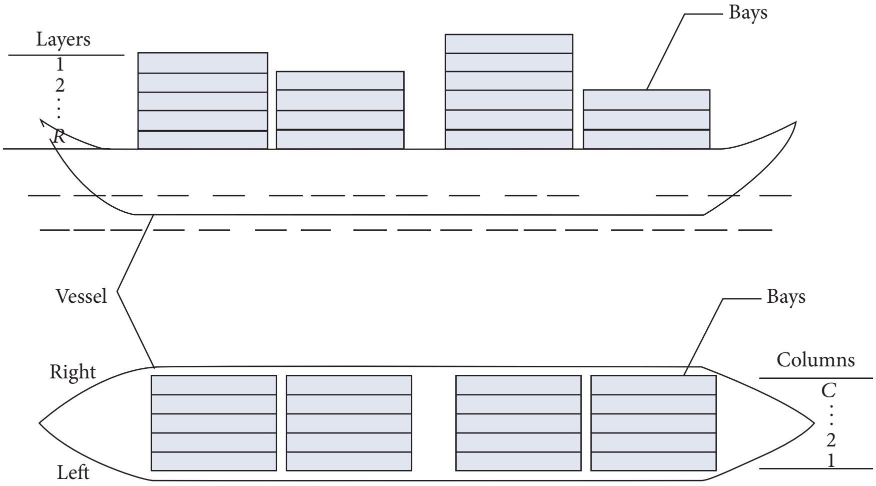


Figure 1: Illustration of a container vessel [3].

In the bay, there are totally N target containers to be loaded and then unloaded in the T ports. Each container i with weight w_i is loaded at port s_i and then unloaded at port d_i. The containers in any stacks are stowed in a way such that heavier containers are in lower layers and lighter containers are in higher layers. Consider any target container i at port d_i such that another nontarget container j is stowed above i in the same stack. As $d_j > d_i$, container j is called a *blocking container*, and it has to be reshuffled once for unloading the target container i. The reshuffle of j causes one unit shifting fee C_{d_j} at the port. Since the unit shifting fee varies in different ports, the stowage planning problem under consideration is to produce a feasible reshuffle schedule such that the total shifting fee in the T ports is minimized. It means that the numbers of shifts in those ports with highest unit shifting fees shall be minimized to some extent. Moreover, due to the load balancing requirement for the whole vessel, it requires load balance in different columns of the bay, and we accordingly assume there is an upper bound on the weight of each stack at any time.

4. Mathematical Formulation

In this section we first give basic notations and then present the objective function and constraints of the MIP model.

4.1. Notations

Indices

t: the index of ports

i, j: the indices of containers

c: the index of container columns or stacks in the bay

r: the index of layers, and $r = 1, R$ for the highest and lowest layers respectively

Input Parameters

T: the number of ports

N: the total number of containers to be loaded/unloaded in the T ports

C: the number of columns or stacks in the bay

R: the number of layers in each stack

w_i: the weight of container i

s_i: the departure port for loading container i

d_i: the destination port for unloading container i

D_t: the set of target containers at port t, that is, $D_t = \{i \mid d_i = t\} \cup \{i \mid s_i = t\}$

U_t: the set of nontarget containers at port t, that is, $U_t = \{i \mid s_i < t < d_i\}$

C_t: the unit shifting fee at port t

W: the maximum allowable weight of containers stowed in each stack

M: a sufficiently large positive integer

Decision Variables

$X^t_{i,c,r}$: equals 1 if container i is located in slot (c, r) when the vessel arrives port t, 0 otherwise

$Y^t_{i,c,r}$: equals 1 if container i is located in slot (c, r) when the vessel leaves port t, 0 otherwise

$Z^t_{i,c,r}$: equals 1 if container i is reshuffled once in slot (c, r) at port t, 0 otherwise

4.2. MIP Formulation

4.2.1. The Objective Function. Each shift of a blocking container in the bay produces one unit of shifting fee C_t at port t, and the objective is to minimize the total shifting fee in all the T ports:

$$\min \quad \sum_{t=1}^{T}\sum_{i=1}^{N} C_t Z^t_{i,c,r}, \quad \forall c \le C,\ r \le R. \tag{1}$$

4.2.2. The Model Constraints. Container i cannot be stowed at any slot (c, r) of the bay either before its departure port s_i or after its destination port d_i:

$$\begin{aligned}&\sum_{c=1}^{C}\sum_{r=1}^{R} Y_{i,c,r}^{t} = 0, \quad \forall i \le N,\ t < s_i,\\ &\sum_{c=1}^{C}\sum_{r=1}^{R} Y_{i,c,r}^{t} = 0, \quad \forall i \le N,\ t \ge d_i.\end{aligned} \tag{2}$$

The total weight of containers in each stack c cannot exceed the upper bound W at any port t:

$$\sum_{i=1}^{N}\sum_{r=1}^{R} w_i Y_{i,c,r}^{t} \le W, \quad \forall c \le C,\ t \le T. \tag{3}$$

After finishing loading/unloading operations at any port t, no more than R containers in each stack or column c are stowed:

$$\sum_{i=1}^{N}\sum_{r=1}^{R} Y_{i,c,r}^{t} \le R, \quad \forall c \le C,\ t \le T. \tag{4}$$

In any stack a heavier container is not allowed to be laid above a lighter container at any port:

$$\begin{aligned}&w_i Y_{i,c,r}^{t} - \sum_{j=1, j\ne i}^{N} w_j Y_{j,c,r+1}^{t} \le 0,\\ &\qquad \forall i \le N,\ c \le C,\ t \le T,\ r \le R-1.\end{aligned} \tag{5}$$

Each slot (c, r) of the bay is stowed with at most one container at any time, and each container is stowed in a single slot during its transportation:

$$\begin{aligned}&\sum_{i=1}^{N} Y_{i,c,r}^{t} \le 1, \quad \forall t \le T,\ c \le C,\ r \le R,\\ &\sum_{c=1}^{C}\sum_{r=1}^{R} Y_{i,c,r}^{t} = 1, \quad \forall i \le N,\ t \in [s_i, d_i - 1].\end{aligned} \tag{6}$$

If container i is not stowed in the bottom of some stack, then there must exist another container stowed beneath it:

$$\begin{aligned}&Y_{i,c,r}^{t} \le \sum_{j=1, j\ne i}^{N} Y_{j,r+1,c}^{t},\\ &\qquad \forall i \le N,\ t \le T,\ c \le C,\ r \in [1, R-1].\end{aligned} \tag{7}$$

The slot locations of containers in the bay may change only at ports during loading/unloading operations, but not on the way between any two ports. That is, the slot status at the time when the vessel is leaving some port is the same as when the vessel arrives the next port. Especially at the last port T, all the remaining containers in the bay are unloaded:

$$\begin{aligned}&X_{i,c,r}^{t+1} = Y_{i,c,r}^{t}, \quad \forall i \le N,\ c \le C,\ r \le R,\ t \le T-1,\\ &Y_{i,c,r}^{T} = 0, \quad \forall i \le N,\ c \le C,\ r \le R.\end{aligned} \tag{8}$$

If a nontarget container i changes its slot location in the bay at port t, then it must be shifted once at the port:

$$Z_{i,c,r}^{t} \ge X_{i,c,r}^{t} - Y_{i,c,r}^{t}, \quad \forall c \le C,\ r \le R,\ t \le T,\ i \in U_t. \tag{9}$$

If a nontarget container i is loaded on the top of a target container j in the same stack at port t. After finishing loading/unloading operations, container i does not change its location, but container i must be shifted at the port:

$$\begin{aligned}&Z_{i,c,r_1}^{t} \ge 1 - M \cdot \left(2 - X_{i,c,r_1}^{t} - X_{j,c,r_2}^{t}\right),\\ &\qquad \forall c \le C,\ t \le T,\ r_1 < r_2,\ i \in U_t,\ j \in D_t.\end{aligned} \tag{10}$$

Nontarget container j is loaded on the top of nontarget container i in one stack at port t. After finishing loading/unloading operations, although container j has not changed its location, it must be shifted once at the port:

$$\begin{aligned}&Z_{j,c,r_1}^{t} \ge X_{j,c,r_1}^{t} + Y_{j,c,r_1}^{t} + X_{i,c,r_2}^{t} + Y_{i,c,r_2}^{t} - 2,\\ &\qquad \forall c \le C,\ t \le T,\ r_1 < r_2,\ i, j \in U_t.\end{aligned} \tag{11}$$

4.2.3. Variable Ranges. The ranges of decision variables are given as follows:

$$\begin{aligned}&X_{i,c,r}^{t}, Y_{i,c,r}^{t}, Z_{i,c,r}^{t} \in \{0, 1\},\\ &\qquad r \le R,\ c \le C,\ t \in T,\ i \le N.\end{aligned} \tag{12}$$

5. Solution Procedure

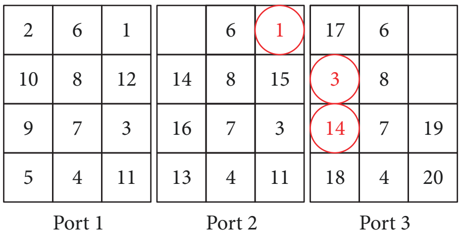
5.1. Optimal Solutions by Cplex for Small Instances. In this section we first consider a small instance where the bay consists of three columns by four layers. That is, there are totally $3 \times 4 = 12$ slots for stowing at most 12 containers at a time in the bay. The vessel visits four ports $(1, 2, 3, 4)$ sequentially. The weight of each container ranges from 1 to 3 where a larger number means a heavier weight. A heavier container is to be stowed in a lower layer of a stack at any time. The maximum loadable weight of the containers in each stack is 8 at any time. Notice that there are no containers in the bay before loading operations at port 1, and all the remaining containers in the bay at port 4 are to be unloaded. It implies that there occurs no shifting fee at either the first port or the last port, while container shifts may happen at ports 2 and 3. The unit shifting fees are 15 and 40 at ports 2 and 3, respectively.

In the instance, there are totally 20 target containers to be loaded and unloaded in the 4 ports. The detailed data of the containers, including their indices, weights, departure, and destination ports, are listed in Table 1. For example, container 1 with weight 1 is to be loaded at port 1 and unloaded at port 3.

We solve the above small instance with Cplex 12.6 solver and the computational results are shown in Figure 2 which illustrates the stowage of the containers in the first three ports. There are four rectangles in each column representing the four layers of each stack, and the digit inside each rectangle denotes the container index. For example, in port 1, there

Table 1: Input data of the containers in the small instance.

Container ID	Weight	Departure port	Destination port
1	1	1	3
2	1	1	2
3	2	1	4
4	2	1	4
5	3	1	2
6	1	1	4
7	1	1	4
8	1	1	4
9	3	1	2
10	1	1	2
11	3	1	3
12	2	1	2
13	3	2	3
14	2	2	4
15	1	2	3
16	3	2	3
17	2	3	4
18	2	3	4
19	1	3	4
20	2	3	4



2	6	1		6	1	17	6	
10	8	12	14	8	15	3	8	
9	7	3	16	7	3	14	7	19
5	4	11	13	4	11	18	4	20
	Port 1			Port 2			Port 3	

Figure 2: An optimal stowage plan of the small instance by Cplex.

are 12 containers to be loaded and stowed at different slots according to their relative weights and their unloading port sequences. During the voyage, only three containers 1, 3, and 14 are shifted at ports 2 and 3, respectively. It results in a total shifting fee of 1 × 15 + 2 × 40 = 95.

For a slightly larger instance where there are five columns by six layers in the bay, 55 target containers are involved in the four ports. The computation time of the solver rises exponentially as the input size increases. Together with the NP-hardness of the considered problem, we propose a genetic algorithm to solve larger instances.

5.2. Genetic Algorithm. As we known, the genetic algorithm has been extensively applied in various applications due to its efficient performance. Its basic procedure is shown in Figure 3. In this paper, we adopt the genetic algorithm to solve large instances of the stowage planning problem. We describe it in detail in the following.

5.2.1. Chromosome Representation. We determine the chromosomal gene segments by the number of ports excluding the last port at which all the remaining containers are unloaded and no shifting fee occurs. That is, there are $T-1$ gene segments. Each chromosome is a solution of the stowage planning.

The length of each gene segment represents the number of containers to be loaded at the corresponding port. Each gene in the gene segment denotes the stack location of a container and is of a value within $0, \ldots, C$, where 0 means there is no container to be loaded at all. For example, the third gene of the first gene segment is equal to 2, representing that the third container is loaded in stack 2 at port 1 (see Figure 4). Since each container is of a different weight, its layer location is accordingly determined by the ranking of its weight in the stack. In this way, it comes to the container's stowage plan.

5.2.2. Population Initialization and Individual Feasibility. We present a distribution mechanism to guarantee that each generated solution is feasible. The mechanism is described as follows. The individual feasibility is verified via the weight limitation of the containers in each stack and the height constraint of the stack. We repeatedly produce a number of individuals and discard the infeasible individuals among them until the number of feasible individuals satisfies the required specific scale.

5.2.3. Fitness Evaluation. We calculate the fitness value of each individual based on the value of the objective function. With the minimization objective, the smaller the objective function value, the better the fitness, and it is more likely to be selected as parents. Calculations of the fitness values are based on the slot position of the containers in the bay.

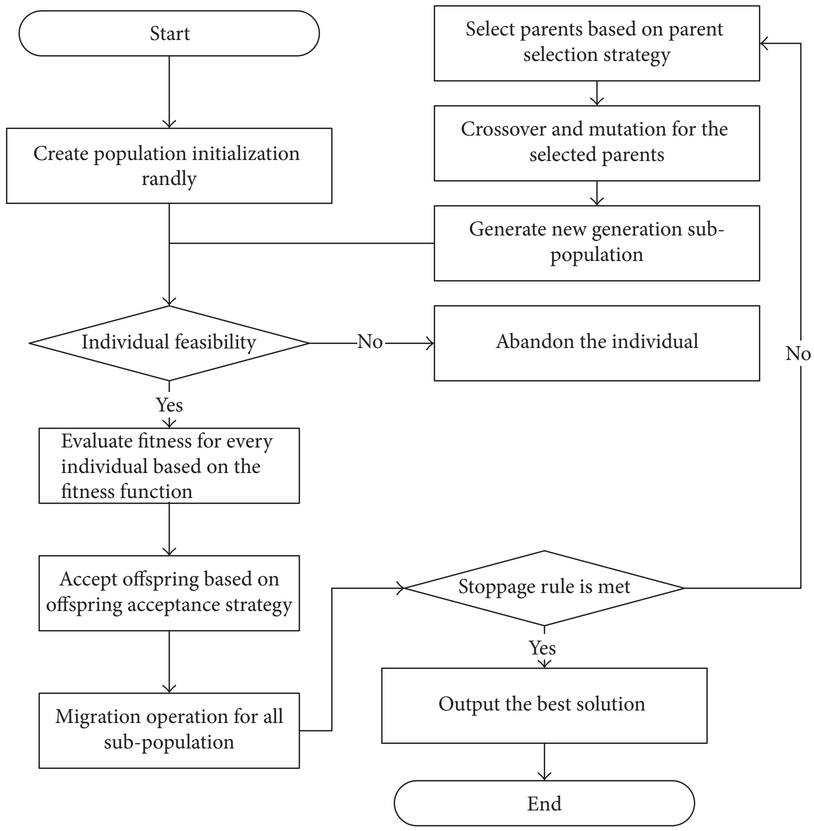
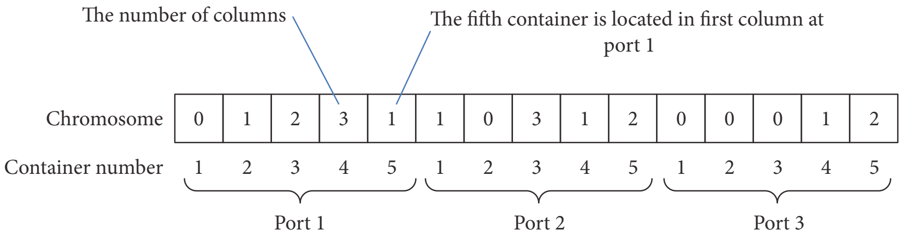

Start
Create population initialization randly
Individual feasibility
No
Abandon the individual
Yes
Evaluate fitness for every individual based on the fitness function
Accept offspring based on offspring acceptance strategy
Migration operation for all sub-population
Stoppage rule is met
Yes
Output the best solution
End
No
Select parents based on parent selection strategy
Crossover and mutation for the selected parents
Generate new generation sub-population

Figure 3: The flowchart of GA algorithm [16].



Chromosome	0	1	2	3	1	1	0	3	1	2	0	0	0	1	2
Container number	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
	Port 1					Port 2					Port 3				

Figure 4: Illustration of chromosome representation.

5.2.4. Crossover and Mutation. There are several typical ways for crossover operation such as single-point crossover, two-point crossover, multipoint crossover, and uniform crossover. In this work, we adopt two-point crossover for crossover operation as follows. Firstly two different crossover points in a gene segment are randomly generated for two chromosomes selected for crossover operation and the value of the crossover points did not exceed the length of chromosome. The digits between the crossover points in the two chromosomes are then exchanged, while keeping the other parts of each chromosome unchanged. After the exchange two new chromosomes are generated (see Figure 5 for an illustration).

The mutation operation of each chromosome occurs randomly with a predetermined probability, which is called the mutation rate. In the mutation, two mutation positions in the chromosome are randomly selected and the two corresponding digits are exchanged, generating a new chromosome. The feasibility of the new chromosome is to be verified in the next step of the algorithm.

5.2.5. Feasibility Verification. The operations of crossover and mutation may result in infeasible solutions or chromosomes. Therefore it is necessary to verify and justify them. The fundamental verifying rules are as follows: (1) the total weight of each stack is no more than W; (2) there are at most R containers in each stack; (3) each container must be stowed in the bay on leaving its departure port and before its destination port; (4) in any stack a heavier container is not allowed to be laid above a lighter container at any port. In each generation, we repeatedly generate new chromosomes and then identify and discard the ones violating any of the above

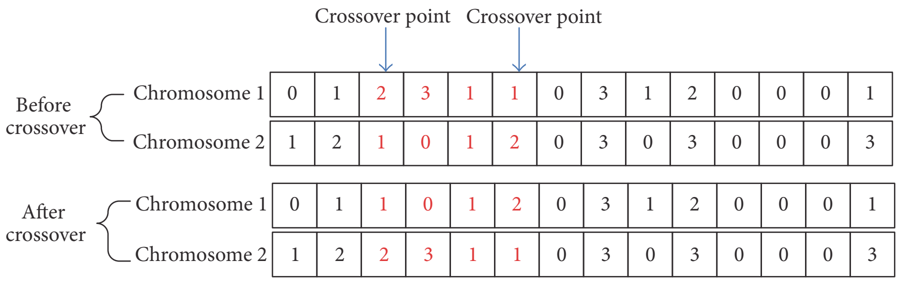


FIGURE 5: Illustration of crossover operation.

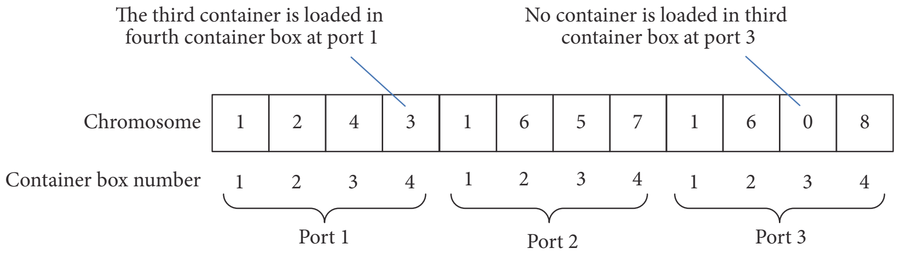

The third container is loaded in fourth container box at port 1
No container is loaded in third container box at port 3
Chromosome
1 2 4 3 1 6 5 7 1 6 0 8
Container box number
1 2 3 4 1 2 3 4 1 2 3 4
Port 1
Port 2
Port 3

FIGURE 6: The new chromosome representation.

rules until the number of offsprings satisfies the population size requirement.

5.3. An Improved Genetic Algorithm

5.3.1. The New Chromosome Representation. We repeatedly produce the feasible individuals until they satisfy the required constraints, but the chromosome length of the first algorithm is too long and the effect of cross-mutation operation on improving the fitness of the initial population is affected. So, the main result is derived from the solution of the initial population generated. In this section, based on the original genetic algorithm, we propose an improved genetic algorithm. We called the original algorithm GAI and the improved one GAII.

The improved chromosome is also composed of $T - 1$ gene segments, the length of each segment is no longer the total number of containers in the route, but the total number of container boxes in the bay. The improved chromosome structure is shown in Figure 6. In the chromosome, there are four container boxes in the bay, and the third container is loaded in fourth container box at port 1. By the chromosome, the third container is unloaded at port 2. Note that if there is no container loaded in a container box, the chromosome gene value is equal to 0. In each gene segment, the containers will be loaded according to their weight at every stack. We repeatedly produce the feasible individuals until satisfying the required scale. Besides, fitness function and cross-mutation operations are the same as in GAI.

5.3.2. Feasibility Verification. After cross-mutation operations, the test rules are as follows: (1) there are no repeated digits in each chromosome segment, except for the number 0; (2) the total weight of each stack is no more than W; (3) each container must be stowed in the bay on leaving its departure port and before its destination port. After cross-mutation operations, if the chromosome does not satisfy all the test rules, then it is discarded.

6. Numerical Experiments

6.1. Performance Comparison of GAI and GAII. As we described in previous section, the stowage plan can be produced by Cplex on small-scale instances. In order to measure the solution quality obtained by GAI and GAII, we first solve the problem by Cplex on small-scale instances. The GA simulation is set up to 100 populations and performed with the limit of 100 generations, coding with Matlab language on a computer (Inter Core i5 CPU, 3.00 GHz; Memory, 4 G). The instances used in the experiment are randomly generated. We assume that the shipping line contains 8 ports and the unit shifting fees in each port are 30, 15, 40, 50, 80, 50, 70, and 25 (RMB), respectively. With different capacities of the bay (C, R) and the number of loaded containers in the route, we test both small and large instances. The total shifting fees and CPU running times in seconds of the solutions produced by Cplex, GAI, and GAII, respectively, on the instances are shown in Table 2.

From the numerical results in Table 2, Cplex can solve the small instances with up to 147 containers efficiently but it cannot output solutions for large instances with 196 or more containers (it is denoted by "/"). The GAI can solve the large-scale instances, while the solutions are of low qualities. GAII can achieve better load planning and especially obtains an optimal solution for the instance with 196 containers. It can effectively solve this kind of stowage problem in a short time, which helps to save operational cost for the shipping company during the route.

Table 2: Computational results of Cplex, GAI, and GAII algorithms.

Total number of containers	(C, R)	Cplex solution	CPU time (s)	GAI solution	CPU time (s)	GAII solution	CPU time (s)
42	$(4, 3)$	0	82	120	280	0	198
70	$(5, 4)$	0	468	280	650	0	324
105	$(6, 5)$	0	2422	1320	968	0	483
147	$(7, 6)$	0	29587	2300	1324	15	673
196	$(8, 7)$	/	/	1870	1726	0	1003
252	$(9, 8)$	/	/	4080	2446	205	1224
315	$(10, 9)$	/	/	4950	2719	335	1466
385	$(11, 10)$	/	/	3745	2882	210	1731

Table 3: Computational results of the GAII for uniform and nonuniform unit shifting fee cases.

Total number of containers	(C, R)	Uniform-shifting amount	Uniform-shifting fees	Nonuniform amount	Nonuniform fees
42	$(4, 3)$	0	0	0	0
70	$(5, 4)$	0	0	0	0
105	$(6, 5)$	0	0	0	0
147	$(7, 6)$	0	0	0	0
196	$(8, 7)$	3	146.25	3	125
252	$(9, 8)$	8	390	13	380
315	$(10, 9)$	17	828.75	17	545
385	$(11, 10)$	20	975	22	650

6.2. Performance Comparison of GAII between Cases with Uniform and Nonuniform Shifting Fees. As previously described, shifting activities in different container ports result in different costs. Besides, minimizing the shifting amounts of container is a special case of minimizing shifting fees. In order to show the influence of shifting fee on stowage planning, we design the numerical experiments on both problems and solve them by GA. We assume that the shipping line contains 8 ports, and again all the instances used in the experiment are randomly generated. For the nonuniform shifting fee case, we assume that the unit shifting fees in the 8 ports are equal to 30, 15, 45, 60, 100, 50, 70, and 20 (RMB), respectively. For the uniform shifting fee case, without loss of generality, we set its unit shifting fee as the average fee over the 8 ports; that is, the uniform unit shifting fee in each port is 48.75 (RMB). The results of GAII for the above two cases on a set of instances are shown in Table 3. "Uniform-shifting amount" and "uniform-shifting fees" represent the total shifting amount and shifting fees, respectively, in the uniform unit shifting fee case. "Nonuniform amount" and "nonuniform fees" denote, respectively, the total shifting amount and shifting fees in the nonuniform unit shifting fee case.

By Table 3, for both cases with uniform and nonuniform unit shifting fees, the GAII optimally solves the first four instances with at most 147 containers. For all the rest four instances with 196 or more containers, we observe that the solutions in the case with nonuniform unit shifting fees correspond to smaller objective values than that in the other case with uniform unit shifting fees. It implies that in the nonuniform unit shifting fee case, most shifting activities are assigned to those ports with the cheapest shifting fees, although it may cause a few more shifts of reshuffle containers. For example, for the last instance with 385 containers, there are $22 - 20 = 2$ more shifts of reshuffling containers but $975 - 650 = 325$ RMB less shifting fee in the nonuniform unit shifting fee case, compared to the uniform unit shifting fee case. Therefore, we conclude that when the shifting fees are nonuniform in different ports, in order to avoid generating a large amount of shifting fee in the ports with expensive shifting fees, shifting activities may happen in some preceding ports with cheaper shifting fees. Although such adjustment may cause more container shifts in total, the total cost of shifting containers can be well reduced.

7. Conclusion

In this paper, we study the stowage planning of a single bay of a vessel in multiple container ports, focus on the case with nonuniform-shifting fees, and establish a mixed integer programming model. The genetic algorithm and improved genetic algorithm are proposed to solve the model. Experiment results show the validity of the model and the proposed genetic algorithms. What is more, by comparing the stowage planning between the case with nonuniform shifting fee and the case with uniform shifting fee, we conclude that a stowage planning in the former case may result in more container shifts but less shifting fees than in the latter case. Since load balance of a vessel is also important in shifting activities and it can be affected by several factors, such as the gravity of the vessel and container types. Considering a variety of constraints about stowage planning is the next focus of our research.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China (71428002, 71432007, 71531011, 71771048, 71571134, and 71571135). The work is also supported by the Fundamental Research Funds for the Central Universities, Shanghai Pujiang program (17PJC046), MOE (Ministry of Education in China) Project of Humanities and Social Sciences (12YJC630284), and Institute of Nonlinear Sciences in Donghua University.

References

[1] H. J. Carlo, I. F. A. Vis, and K. J. Roodbergen, "Transport operations in container terminals: Literature overview, trends, research directions and classification scheme," *European Journal of Operational Research*, vol. 236, no. 1, pp. 1–13, 2014.

[2] M. Avriel, M. Penn, and N. Shpirer, "Container ship stowage problem: Complexity and connection to the coloring of circle graphs," *Discrete Applied Mathematics*, vol. 45, pp. 271–279, 1998.

[3] D. Ambrosino, A. Sciomachen, and E. Tanfani, "Stowing a containership: The master bay plan problem," *Transportation Research Part A: Policy and Practice*, vol. 38, no. 2, pp. 81–99, 2004.

[4] D. Ambrosino, M. Paolucci, and A. Sciomachen, "A MIP heuristic for multi port stowage planning," *Transportation Research Procedia*, vol. 10, pp. 725–734, 2015.

[5] F. Parreño, D. Pacino, and R. Alvarez-Valdes, "A GRASP algorithm for the container stowage slot planning problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 94, pp. 141–157, 2016.

[6] L. Wei, W. Zhu, and A. Lim, "A goal-driven prototype column generation strategy for the multiple container loading cost minimization problem," *European Journal of Operational Research*, vol. 241, no. 1, pp. 39–49, 2015.

[7] A. Delgado, R. M. Jensen, K. Janstrup, T. H. Rose, and K. H. Andersen, "A constraint programming model for fast optimal stowage of container vessel bays," *European Journal of Operational Research*, vol. 220, no. 1, pp. 251–261, 2012.

[8] M. F. Monaco, M. Sammarra, and G. Sorrentino, "The terminal-oriented ship stowage planning problem," *European Journal of Operational Research*, vol. 239, no. 1, pp. 256–265, 2014.

[9] D. Ambrosino, D. Anghinolfi, M. Paolucci, and A. Sciomachen, "A new three-step heuristic for the Master Bay Plan Problem," *Maritime Economics & Logistics*, vol. 11, no. 1, pp. 98–120, 2009.

[10] M. Liu, F. Chu, Z. Zhang, and C. B. Chu, "A polynomial-time heuristic for the quay crane double-cycling problem with internal-reshuffling operations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 81, pp. 52–74, 2015.

[11] F. Meisel and M. Wichmann, "Container sequencing for quay cranes with internal reshuffles," *OR Spectrum*, vol. 32, no. 3, pp. 569–591, 2010.

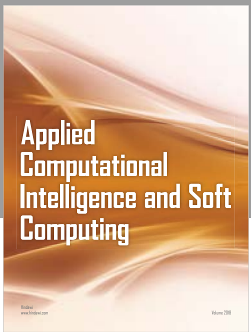
[12] E. J. Araújo, A. A. Chaves, L. L. De Salles Neto, and A. T. De Azevedo, "Pareto clustering search applied for 3D container ship loading plan problem," *Expert Systems with Applications*, vol. 44, no. 1, pp. 50–57, 2015.

[13] D. Ding and M. C. Chou, "Stowage planning for container ships: A heuristic algorithm to reduce the number of shifts," *European Journal of Operational Research*, vol. 246, no. 1, pp. 242–249, 2015.

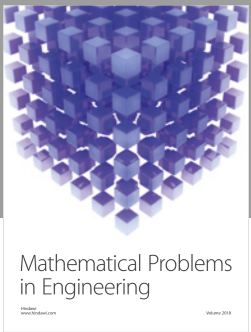
[14] S. Wang, "Optimal sequence of container ships in a string," *European Journal of Operational Research*, vol. 246, no. 3, pp. 850–857, 2015.

[15] A. G. Ramos, J. F. Oliveira, J. F. Gonçalves, and M. P. Lopes, "Dynamic stability metrics for the container loading problem," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 480–497, 2015.

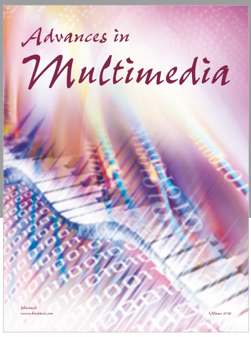
[16] D. Chang, Z. Jiang, W. Yan, and J. He, "Integrating berth allocation and quay crane assignments," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 6, pp. 975–990, 2010.

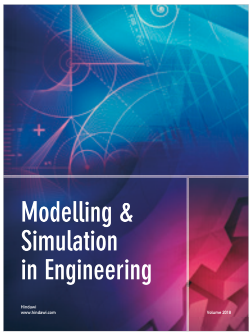

Applied Computational Intelligence and Soft Computing

The Scientific World Journal


Mathematical Problems in Engineering

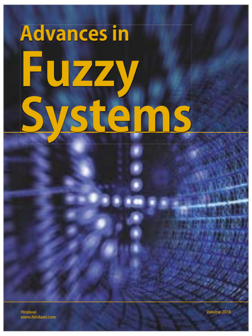
Journal of Engineering

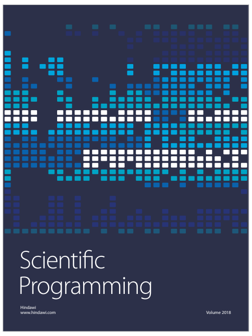

Advances in Multimedia

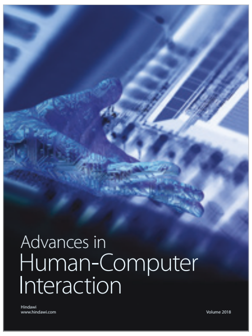

Modelling & Simulation in Engineering

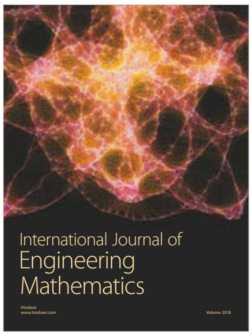
Advances in Artificial Intelligence

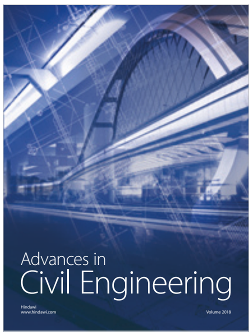
International Journal of Reconfigurable Computing

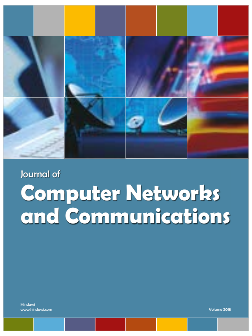

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

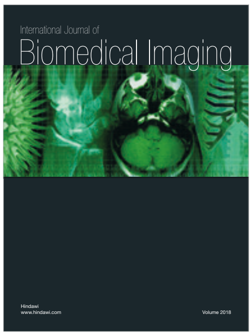

International Journal of Engineering Mathematics


Advances in Civil Engineering

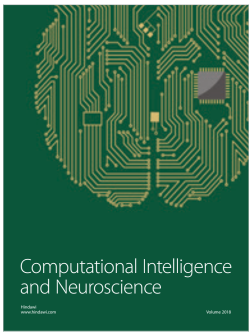

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience