ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

Container Stowage Plan based on Harmony-inspired Genetic Algorithm

Qichen Ding

Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China

*15869104690@163.com

Abstract

In order to improve the efficiency of ship loading and unloading, the optimization goal is to minimize the amount of box dumping on the entire track, and the water planning model that takes into account the stability of the aircraft. A Harmony-Inspired Genetic Algorithm (HIGA) is designed to solve the stowage problem, and parameters are introduced to determine whether the current optimal algorithm is local optimal or optimal, so as to call the harmony. Search for optimization algorithms, and dynamically update the maximum number of iterations of the algorithm. Time-consuming, and with the help of the local optimization of the harmonic search speed, the solution time is reduced by 21.75%, the average number of ships is reduced by 26.87%, and the solution efficiency and the optimal solution are improved. For the quality of container ships, look at the pre-allocation plan. Ready to provide a certain reference.

Keywords

Container Stowage; Route Stowage Planning; Integer Programming; Harmony Search; Genetic Algorithm.

1. Introduction

For a port, shorter docking time means higher efficiency and competitiveness of port trade environment. High quality container stowage scheme can improve the loading and unloading efficiency of the terminal, reduce the docking time of the ship, and thus save the operating cost of the ship. Container stowage, as a core link in container transportation, mainly solves the problem of container loading when ships call at various ports, reduces the operation of reverse container in the midway port, and improves the work efficiency of port and ship. In the formulation of container stowing plan, a variety of objectives need to be considered, such as cabin space utilization, cargo safety, ship stability, berthing time in port, etc. The model is complex, and a lot of calculations are involved in the analysis process. The container stowage planning problem has been proved by Avrie [1] to be an NP-hard problem. Shen [2] refined the relevant stowage feature vectors and applied the DQL method to solve the stowage problem of container ships, thus improving the solving efficiency.

All manuscripts must be in English, also the table and figure texts, otherwise we cannot publish your paper. Please keep a second copy of your manuscript in your office. When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question. Should authors use tables or figures from other Publications, they must ask the corresponding publishers to grant them the right to publish this material in their paper. Use italic for emphasizing a word or phrase. Do not use boldface typing or capital letters except for section headings (cf. remarks on section headings, below). Korach [3] solved the stowage problem in two stages. First, a heuristic algorithm was designed to solve the master bay plan, and then a large

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

neighborhood search algorithm was used to solve the slot plan. Zheng feifeng [4,5] took the loading balance on both sides of the bay as a security constraint and established a multi-port stow model for a single bay. Yang Wei [6] took ship's transverse stability as navigation restriction condition, constructed a stowing model and solved it with adaptive genetic algorithm. Liu Zhixiong [7] optimized the stowing scheme by adding rules based on torque balance and column by column into the hybrid evolution strategy. Jin Jian [8] et al., aiming at the problem of container cabin stowage planning in the loading and unloading process, comprehensively considered the influence of the number of container turning in the yard and the number of terminal loading and unloading equipment movement on terminal operation efficiency, constructed a multi-objective optimization model and designed monte Carlo tree search algorithm to solve the problem. Li Jun et al. [9] aimed at minimizing the number of stack occupancy and designed a heuristic algorithm to solve the loading scheme. Tian Wei et al. [10] designed a three-stage heuristic algorithm to solve the ship staowaging sorting problem by building a model with the goal of the shortest loading time and the least number of container turning in the yard. Zhu Huiling et al. [11] on the premise of meeting the ship's stowage plan, proposed the optimal stack dumping strategy based on the traditional shortest stack and adjacent stack strategies, respectively established the loading sequence models under three kinds of dumping strategies with the goal of minimizing the amount of dumping, and designed the genetic algorithm to optimize and solve the models. Savas [12] took the minimum number of inverted containers as the optimization goal, proposed an integer programming model based on two-stage heuristic algorithm to obtain the lower bound solution as the initial solution of the problem, and further solved the optimal solution of the problem by switching heuristic algorithm. Li Jun et al. [13] constructed a single-port single-stage stowage decision model based on the rolling scheduling strategy, and then realized the dynamic stowage decision of liner routes by rolling. Yang Ben et al. [14] proposed a depth-first and dynamic depth-multi-branch search stowing algorithm based on the stowing task arranged by the bridge. Xu Jin et al. [15] proposed a granularity division strategy to divide ship shell structure into multiple storage units, and constructed a whole-route stowage model of inland river container ships with the goal of minimizing the occupancy rate of ship storage units.

When studying container stowage, domestic and foreign literatures all aim at reducing terminal operation time. Some literatures take single port stowage as the research object to improve terminal operation efficiency by reducing the number of container turning in the yard, the movement of loading and unloading equipment and optimizing the order of container dispatch. Some literatures take benin stowage as the research object and improve the overall operating efficiency of wharves by minimizing the total dumping times of shipping lines, but few studies comprehensively consider the restrictions on ship navigation safety. In this paper, the ship's high transverse stability is taken as the navigation safety limit, and the optimization model of the whole route stow is built with the minimum amount of empty container as the goal. The harmony genetic algorithm is designed to solve the problem, which makes up the defect that the algorithm is easy to fall into the local optimal.

2. Problerm Definition

Full-line stowage of container ships refers to the formulation of stowage plans for ships by stacking containers in fixed places in accordance with certain packing rules and taking into account the transportation requirements of various ports under the condition of ensuring safe navigation of ships. When container ships visit ports up to a certain amounts, it is inevitable that there will be backpacking operations, Unloading and loading of the same container in the same port can be caused by two reasons. As shown in the left column of the stack in figure.1, when arriving at port 4, the container at port 4 in this column needs to be unloaded, and the four containers loaded at port 5 must be unloaded first. We say that there have 4 shift operations at this time. Another non-mandatory shift, as shown in figure 1 stack, as shown in the right column in the port 4, there are four destination for 6 containers for shipment, in order to avoid the port of destination for 6 container loading at the port of destination for above 5 container, two at the bottom of the column must be the port of destination for 5 containers, It was then loaded on top of three containers at port 6 of destination, and 2 shift operations in the

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

process happened. Ship stability is the ability of a ship to maintain its original position after the disappearance of external forces, and it is one of the main indicators to measure the safety of ship navigation. Under the condition of a certain displacement, when the ship produces a small roll, the higher the transverse stability is, the stronger the ability to resist the tilting moment is. High transverse stability can be expressed as $GM = KM - KG$, Where KM is the distance between the ship's transverse metacenter and the baseline, which can be obtained by checking the hydrostatic curve. KG is the distance between the ship's center of gravity and the baseline, $KG = (\sum_m H_m w^m) / \Delta T_i$, Where H_m represents the distance between the ship and the vertical center of gravity of container m, and ΔT represents the displacement of the ship when it leaves port i.

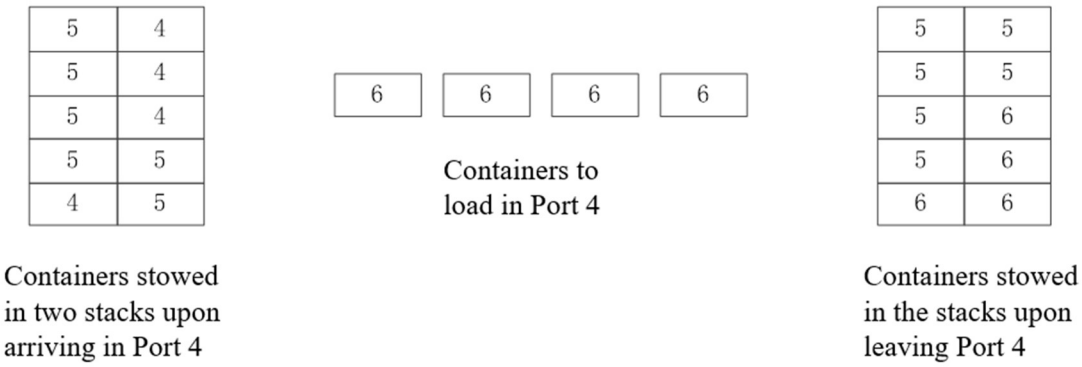

Figure 1. An example of mandatory and nonmandatory shifts

3. Mathematical Model

3.1 Basic Assumptions

(1) Only the 20ft and 40ft standard containers are considered, and the two containers cannot be mixed up and down.(2) Each bay contains the same number of stacks with the same number of stack layers.(3) The container transport information of each port is known, and the container ship is empty before loading at the first port and after unloading at the last port.

3.2 Related Symbols

Ports noted by 1,2,... , n. Related parameters are as follows:

I :container set at origin port, $i = \{1, 2, ..., n-1\} \in I$;

J : containers set at destination port, $j = \{2, 3, ..., n\} \in J$;

T_i :the transportation demand of port i;

B:bay set, $b \in B$;

C:set of stack of the bay;

R:number of highest tier of the bay, $r \le R$,rIndicates the tier of the bay;

M :set of all containers to be loaded, The containers are numbered in order according to the port of origin, port of destination and weight from smallest to largest, and marked as:m=(i,j,w), $m \in M$;

M_i :set of the containers have been loaded When the ship leaves port i, M_i^T Represents the set of 20ft containers, M_i^F Represents the set of 40ft containers, $M_i^T \cup M_i^F = M_i$;

w^m : weight of container m;

ΔV : Represents the displacement of the ship when it is empty;

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

ΔT_i: Represents the displacement of the ship when it leaves port i;

H: Represents the height of a standard container.

ξ_{ibcr} :When the ship leaves port i, the container in tier r of stack c of bay b is unloaded at port i +1, then it is 1, otherwise it is 0.

x_{ibcr}^m : When the ship leaves port i, container m loaded in tier r of stack c of bay b, then it is 1; otherwise, it is 0.

The model is as follows:

$$\min \sum_{i=1}^{n-1}\sum_{b\in B}\sum_{c\in C}\sum_{r\in R}\xi_{ibcr} \tag{1}$$

$$\sum_{m\in M} x_{ibcr}^m \le 1; i\in P_1^{n-1}, b\in B, c\in C, r\in R \tag{2}$$

$$\sum_{m\in M_d} x_{icbr}^m = 0; d\ne i\in P_1^{n-1}, b\in B, c\in C, r\in R \tag{3}$$

$$\sum_{b\in B}\sum_{c\in C}\sum_{r\in R} x_{ibcr}^m \le 1; i\in P_1^{n-1}, m\in M \tag{4}$$

$$\sum_{m\in M_i} x_{ibcr}^m \ge \sum_{m\in M_i} x_{ibc,r+1}^m; i\in P_1^{n-1}, b\in B, c\in C, r\in R \tag{5}$$

$$\sum_{m\in M_i^F} x_{ibcr}^m + \sum_{m\in M_i} x_{i,b+1,cr}^m \le 1; i\in P_1^{n-1}, b \text{ is an odd number}, c\in C, r\in R \tag{6}$$

$$\sum_{m\in M_i^F} x_{ibcr}^m + \sum_{m\in M_i} x_{i,b-1,cr}^m \le 1; i\in P_1^{n-1}, b \text{ is an even number}, c\in C, r\in R \tag{7}$$

$$\sum_{m\in M_i^F} x_{ibcr}^m + \sum_{m\in M_i^T} x_{ibc,r+1}^m \le 1; i\in P_1^{n-1}, b\in B, c\in C, r\in R \tag{8}$$

$$\sum_{m\in M_i^F} x_{ibcr}^m + \sum_{m\in M_i^T} x_{ibc,r-1}^m \le 1; i\in P_1^{n-1}, b\in B, c\in C, r\in R \tag{9}$$

$$\xi_{ibcr} \ge x_{ibcr}^m - x_{i+1,bcr}^m; m\in M, i\in P_1^{n-1}, b\in B, c\in C, r\in R \tag{10}$$

$$\xi_{ibcr} \le \xi_{ibc,r+1}; i\in P_1^{n-1}, b\in B, c\in C, r\in R \tag{11}$$

$$\sum_{b\in B}\sum_{c\in C}\sum_{r\in R}\sum_{m\in M_i} x_{ibcr}^m = T_i; i\in P_1^{n-1} \tag{12}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

$$\Delta T_i = \Delta V + \sum_{b\in B}\sum_{c\in C}\sum_{r\in R}\sum_{m\in M} x_{ibcr}^m ; i \in P_i^{n-1} \tag{13}$$

$$GM_i = g(\Delta T_i) - h\frac{\sum_{b\in B}\sum_{c\in C}\sum_{r\in R}\sum_{m\in M} w^m x_{ibcr}^m (r-0.5)}{\Delta T_i} > l; i \in P_1^{n-1} \tag{14}$$

$$x_{ibcr}^m = \{0,1\}; i \in P_1^{n-1}, b \in B, c \in C, r \in R, m \in M \tag{15}$$

$$\xi_{ibcr} = \{0,1\}; i \in P_1^{n-1}, b \in B, c \in C, r \in R \tag{16}$$

(1) is the objective function that minimises the total container shifts on the travle;Constraint (2) defines that no more than one container can be placed in any space on board; Constraint (3) say that the ship leaves port i with only the current port container; Constraint (4) defines that a container can occupy only one space when a ship leaves port i; Constraint (5) defines containers cannot be dangling placed;Constraint (6)-(7) say that if the odd number of shells is occupied, the adjacent even number of shells does not exist. If an even number of shells is occupied, the adjacent odd number of shells does not exist; Constraint (8)-(9) define that 20ft and 40ft containers cannot be mixed up in a same stack; Constraint (10) defines the variable ξ_{ibcr}; Constraint (11) indicates that if the containers on tire r are unloaded, all the containers above it should be unloaded; Constraint (12) indicates that the stowage scheme of the ship at port I must meet the transportation demand; Constraints (13)-(14) are restrictions on navigation safety of ships; Constraints (15)-(16) defines the variables.

In the actual loading work, the linear programming model is difficult to solve because of the large loading scale, and the traditional genetic algorithm has low solving accuracy, which can not meet the actual demand. Therefore, in this paper, harmony algorithm is used to improve genetic algorithm, and Harmony-Inspire Genetic algorithm (HIGA) is proposed to solve the whole route stowing problem.

4. Harmony-inspire Genetic algorithm

4.1 HIGA Algorithm

Harmony search has a strong ability to search, fast convergence rate, the characteristics of few adjustable parameters, so the harmony algorithm combined with genetic algorithm, according to the change of the optimal individual fitness in each generation to judge whether the current solution into a local optimal solution, and then through the harmony search algorithm to optimize the current population, and jump out of local optimal solution, improve the efficiency of iteration. Based on this, the following parameters are introduced:

leader:The best individual in a contemporary population; leaderi+1=leaderi;

trace:The fitness changes of population leader in each generation; if trace=2, it indicates that the fitness value of the optimal individual is at the same level for two consecutive iterations;

trials: The times that harmony algorithm failed to optimize the population and improve the fitness of the optimal individual;

trace_limit: value of the threshold of value. When trace reaches the threshold, it indicates that the algorithm stops in the local optimal region;

HIGA algorithm is shown in Algorithm 1. In the first stage, genetic algorithm is used to generate offspring population. In the second stage, the optimal individual of the offspring is saved, and the fitness value of the optimal individual of the parent is used to judge whether the genetic algorithm falls into local optimum. When the parameter Trace reaches the threshold, phases three and four are entered. In the third stage, HIGA uses harmony algorithm to search for the optimal solution of higher

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

quality in the contemporary population and update the contemporary population. In the fourth stage, HIGA iterations are updated according to the optimization effect of harmony algorithm on the current population.

Algorithm 1: HIGA algorithm.

Input Population size: population_size; Maximum number of iterations of genetic algorithm: max_Iterations; Crossover probability: Pc; Variation probability: Pm; Iterations of harmony algorithm: HS_Iterations; Parameters: trace_limit, trials_limit.

Output Optimal unit of generation ith: leader (GAi).

```
Initialize algorithm specific parameters.
Generate initial generation HIGA0 randomly.
DO.
  GAi=GAEvolution(GAi-1).
  If Leader(GAi)==Leader(GAi-1).
    trace=trace+1.
  End if.
  If trace>=trace_limit.
    trace=0.
    HS=HarmonySearch(GAi,hs_iterations).
    HIGAi=ExtractBestIndividuals(GAi,HS).
    If Leader(GAi)<Leader(HS).
      trials=trials+1.
      If trials>=trials_limit.
        i=i+(hs_iterations/2).
        trials=0.
      End if.
    End if.
    i=i+hs_iterations.
  End if.
  i=i+1.
While(i<GA_iterations).
Return Leader(GAi).
```

4.2 GA Design

4.2.1 Chromosome Encoding

According to the characteristics of full-route stowing, the chromosomes are divided into N-1 genomes, each genome represents the stowing scheme of the corresponding port, and the gene fragments of each genome correspond to the specific container position on the ship. The first two gene fragments of the chromosome respectively represent the first and second layers in the first column of ship 1 shell, and so on. The container number m=(I,j,w) is used as the genetic coding form to represent the information of containers placed in corresponding positions, where the slot code of unplaced containers is m=(0,0,0), and the chromosomal coding is shown in figure.2.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

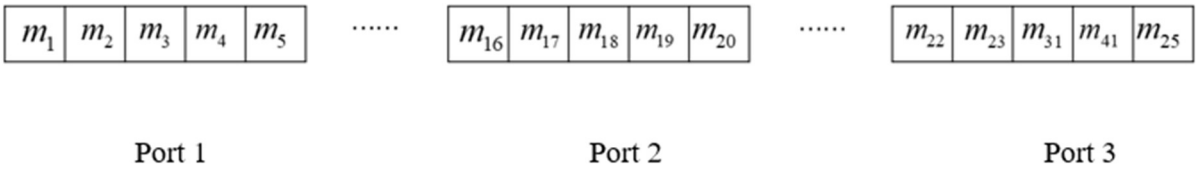

Figure 2. Chromosome encoding

4.2.2 Initial Population

To ensure the quality of the initial solution, the initial solution is generated based on the following rules:

a) Determine the specific loading position of each container when the ship leaves the port 1 without reversing the container;

b) The loading and unloading operation of the ship from port 2 to port n-1 is simulated. At port j, all the containers stacked above the port of destination j are unloaded, and all the containers that need to be re-loaded at port j are loaded from far to near, heavy first and light later.

4.2.3 Genetic Operations

In this paper, the method of roulette was adopted to screen paternal individuals. The greater the individual fitness, the greater the probability of being selected as paternal individuals.

In the genetic process, two-point crossover strategy is adopted to allow only the same genome to carry out crossover operation, that is, crossover operation can only be carried out in the same port. Mutation operation is to randomly generate two mutation points in the same genome and exchange their container positions.

The leader in the parent generation replaces the inferior individual in the child generation to ensure that the high-quality individual is not destroyed and accelerate the convergence of the algorithm.

4.3 Optimization Population by Harmony Search

The stowed plan of the whole route is taken as a set of harmonies, and the stowed state when the ship leaves the port is taken as a sum tone. Figure.3 represents a set of harmonies containing five ports, and Pi represents the i-th sum tone, corresponding to the i-th genome of genetic algorithm chromosome. In order to make the algorithm jump out of local optimum faster and expand the search space of solution, in the harmonic tuning stage, the solution is adjusted by exchanging the position of containers. If the container position at the current port changes, subsequent ports should also make corresponding position adjustment when fine-tuning the harmonization sound.

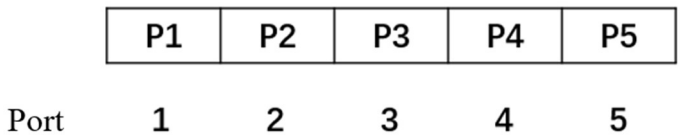

Figure 3. Harmony search encoding

The optimization steps of Harmony Search are shown in Algorithm 2. Memory represents the harmony memory library, HMCR represents the value probability of the Memory library, PAR represents the probability of tune adjustment, BW represents the tune adjustment range, that is, the number of adjustment of current port container positions, and HS_Iterations represents the number of iterations of the harmony algorithm. Evaluation function is the same as fitness function.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

Agrithom2: Harmony search.

Input population: GA_generation; Number of current iterations of genetic algorithm: Iterations; Maximum number of iterations of genetic algorithm: GA_iterations; Harmonic memory bank value probability: HCMR; Harmonic fine-tuning probability: PAR; Harmonic adjustment: BW.

Output Harmonic Memory: new HS Memory.

```
Memory=GA_generations.
Do.
    Generation new candidate X^NEW .
    Foreach ith component in candidate X^NEW .
            If rand(0,1)<HCMR.
              Pick X^NEW from Memory.
              If rand(0,1)<PAR.
                X^NEW =adjust( X^NEW ).
            End if.
            Else.
                Generate X^NEW randomly based on stowage rulers.
            End if.
    End Foreach.
    Calculate fitness of candidate X^NEW .
    If fitness of X^NEW is better than the fitness of worst candidate X_worst from Memory.
        Replace X_worst with X^NEW in the Memory.
    End if.
While(i<iterations).
Return last memory as new HS generation.
```

After using the HS algorithm to optimize the population, the individual fitness of HS and GA groups was compared, and the high-quality individuals were retained as the offspring population of HIGA. The extraction process was shown in Algorithm 3.

Agrithom3: Extraction of optimal population individuals.

Input HIGA algorithm current population: HIGA_generations; Harmony algorithm Memory: HS Memory;

Output HIGA algorithm population after optimization: HIGA_generations.

```
Sort both generations by fitness.
i=1;j=1.
Do.
    If HIGA_generations(i)>HS Memory (j).
        HIGA_generations (i)= HS Memory (j).
        j=j+1.
    End if.
    i=i+1.
While(i<population_size).
Return HIGA_generations.
```

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

4.4 Fitness Function

For the navigation safety of ships, in order to avoid excessive stability height, a standard value is taken, the closer it is to the standard value, the greater the fitness. For the number of unloading operations, the less the number of unloading operations, the greater the fitness. The fitness function is as follows:

$$f = 1/(w_1 * obj1) + 1/(w_2 * obj2) \tag{17}$$

w_1 is the shift weight coefficient, obj1 is the amounts of shifts, w_2 is the stability weight coefficient, obj2 is the stability calculation result.

5. The Example Analysis

5.1 HIGA Calculation Performance Analysis

Design two types of small ships, the structure as shown in Table 1. n represents the number of ports, gap represents the deviation between HIGA result and CPLEX solution. "-" indicates that the model cannot obtain the result within 600s.

Table 1. Solution of example

case	Bay	Stack	Tier	n	Number of container	CPLEX		HIGA		
						Opt	T/s	Opt	T/s	gap/%
1	6	4	3	3	51	51	7.40	51	11.85	0
2	6	4	3	4	65	65	45.13	65	20.82	0
3	6	4	3	5	76	80	167.2	80	41.71	0
4	6	8	4	3	187	187	331.51	187	28.23	0
5	6	8	4	4	179	182	597.12	187	58.24	2.75
6	6	8	4	5	267	-	600	272	100.14	-

For small cases, it can be seen from Table 1 that: In terms of solution quality, HIGA solution results are consistent with CPLEX except for example 5. In terms of solving efficiency, HIGA can obtain high quality solutions in a short time.

Table 2. Basic information of the vessel

Parameter	value
Bay(b)	20
Stack(s)	10
Tier(t)	8
Slot	252
TEU	1600
Molded breadth/m	32.294
Moulded depth/m	21.5
Length between perpendiculars/m	264.2
Weight/t	12917

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

Assumed that the ship calls at five ports in one voyage. The ship information is shown in Table 2, and the initial stability GM of the ship is not less than 0.65m, 20ft container weight is $\{w_1, w_2, w_3\} = \{15t, 20t, 25t\}$,40ft container weight is $\{w_3, w_4, w_5\} = \{30t, 35t, 40t\}$.

Table 3 lists the related parameters. The optimal individuals of the genetic algorithm usually maintain the same fitness in the 35-60 generations. It is found in the experiment that if the value of trace_limit is too low, HIGA's solution results are biased to harmony algorithm. If the value is too high, the performance of HIGA is similar to that of genetic algorithm. For the stowing problem proposed in this paper, when trace_limit is 15, the algorithm shows good performance. Trials_limit determines whether HIGA algorithm reduces the number of iterations. If the parameter value is too low, HIGA may converge prematurely. If the value is too high, the optimization effect of HIGA is not significantly affected. The value of trials_limit must be greater than 3 and smaller than the ratio of trace_limit to GA_iterations.

Table 3. Setting of parameters

Parameters	Value
GA_iterations	500
Population_size	100
Pc	0.9
Pm	0.05
HMCR	0.95
$[PAR_{\min} \quad PAR_{\max}]$	[0.01,0.99]
$[BW_{\min} \quad BW_{\max}]$	[3,800]
Hs_iterations	30
Trace_limit	15
Trials_limit	5

To further verify the validity of HIGA, 10 groups of examples are randomly generated as follows: maximum number of bay occupied by a 40ft container $BF = \left\lfloor \frac{b}{2} * U[0,1] \right\rfloor$, maximum number of bay occupied by a 20ft container $BT = b - 2 * BF$, RT is the number of 20ft remaining container slots, and the number of each weight container $WT_i = \lfloor RT * U[0.1] \rfloor$, RF is the number of 40ft remaining container slots and the number of each weight container $WF_i = \lfloor RF * U[0.1] \rfloor$. GA, HS and HIGA were used to solve the problem, and the results are shown in Table 4. Compared with GA, HIGA saves 21.75% of solving time on average and 26.87% of unloading times on average. Although HS has a short solution time, its solution quality is far inferior to GA and HIGA. HIGA combines the advantages of GA and HS and makes up for their disadvantages in terms of solving time and quality of optimal solution, which further proves the effectiveness of the algorithm.

Volume 8 Issue 4, 2022
DOI: 10.6919/ICJE.202204_8(4).0058

Table 4. Optimal solution

case	TEU	GA		HS		HIGA	
		Opt	Time/s	Opt	Time/s	Opt	Time/s
1	3059	3078	783.21	3243	378.62	3078	582.37
2	3012	3089	695.13	3186	377.25	3067	578.85
3	2985	3077	801.20	3241	503.50	3068	598.07
4	3064	3078	768.56	3188	493.66	3076	574.68
5	3134	3154	743.91	3259	374.62	3160	575.71
6	2897	2974	865.55	3017	379.47	2912	714.45
7	2943	3018	699.86	3124	381.96	3006	567.56
8	2988	3105	703.45	3186	372.11	3105	602.18
9	2741	2853	739.84	3067	391.55	2811	562.74
10	2864	2930	768.33	3152	372.51	2895	565.82
mean value	2969	3036	756.90	3166	402.53	3018	592.24

5.2 HIGA Optimization Performance Analysis

It can be seen from figure 4 that GA algorithm stops only when it reaches the maximum number of iterations, and in the iteration process, the fitness of the optimal individual still does not improve after several iterations. HIGA algorithm determines that when the genetic algorithm falls into the local optimal solution according to the parameters trace and trace_limit, it calls the harmony algorithm to help it jump out of the local optimal solution quickly. Even if it fails, the diversity and adaptability of the population are increased by updating the population, and more excellent individuals are generated in subsequent iterations. When trials reaches the threshold, HIGA updates the number of iterations and obtains feasible solutions superior to GA in shorter iteration time.

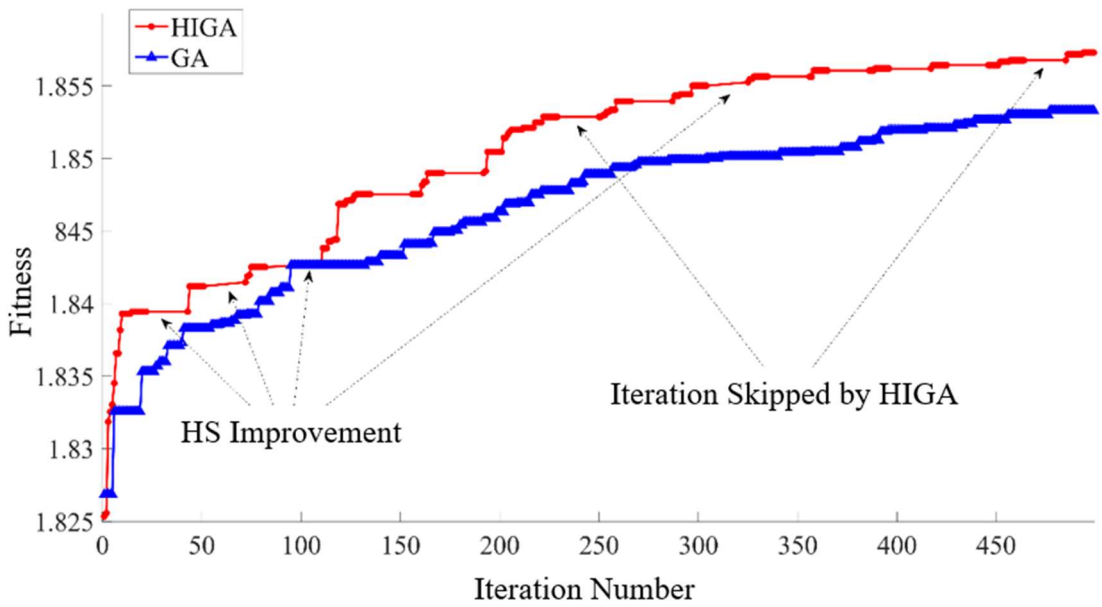

Figure 4. HIGA VS GA comparison based on fitness vs generation

ISSN: 2414-1895 DOI: 10.6919/ICJE.202204_8(4).0058

When the number of docking ports increases, with the size of ship stowed increases, the solution will be more complicated. It can be seen from Figure 5 that there is a significant gap between the solution results of HS algorithm and HIGA algorithm. When the number of ports is less than 5, the results of GA and HIGA are similar. When the number of ports is more than 5, HIGA gradually shows better global search ability than GA.

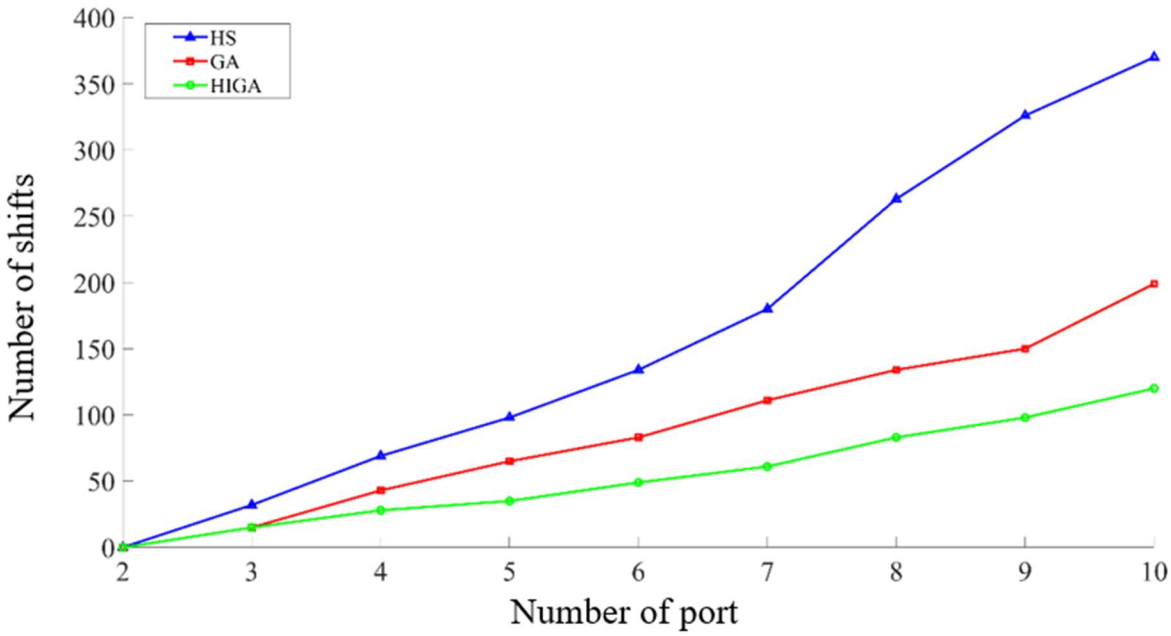

Figure 5. Influence of port number on algorithm performance

6. Conclusion

The main contributions of this paper are as follows:(1) An integer programming model was established based on the actual ship stowage rules and navigation safety constraints, considering the two types of 20ft and 40ft boxes with the minimum amount of dumping as the optimization objective.(2) In this paper, a harmony-genetic algorithm is proposed, which determines whether to call harmony algorithm to optimize the population according to the evolution of individuals in the iterative process, and dynamically updates the iteration times of the algorithm, so as to effectively avoid the local convergence of algorithm optimization and improve the efficiency of iteration.(3) By comparing the optimal solutions of GA, HS and HIGA, it is found that HIGA saves 21.75% of solving time on average and 26.87% of unloading times on average compared with GA. Although HS has a short solution time, its solution quality is far inferior to GA and HIGA. In addition, when the number of affiliated ports is less than 5, the solution results of GA and HIGA are similar. With the increase of the number of ports, HIGA gradually shows better global search ability than GA. HIGA combines the advantages of GA and HS and makes up for their shortcomings in terms of solving time and quality of optimal solution, which further proves the effectiveness of the algorithm and has certain guiding significance for the formulation of multi-port container stowage plan.

References

[1] Avriel Mordecai, Penn Michal. Exact and approximate solutions of the container ship stowage problem. 1993, 25(1-4):271-274.

[2] Y. F. Shen,N. Zhao,M. J. Xia,X. Q. Du. A Deep Q-Learning Network for Ship Stowage Planning Problem[J]. Polish Maritime Research,2017,24(s1).

[3] Korach A, Brouer B D, Jensen R M. Matheuristics for slot planning of container vessel bays[J]. European Journal of Operational Research, 2020, 282(3): 873-885.

[4] F. F. Zheng, Q. H. Mei, M. Liu, et al. A Research on Bi-Objective Exact Algorithm of Stowage Planning for Multiple Container Ports[J]. Industrial Engineering Journal,2018,21(6):1-6.

[5] F. F. Zheng, Q. H. Mei, M. Liu, et al. Research on Genetic Algorithm and Greedy Method of Stowage Planning in Multiple Ports[J]. Operations Research and Management Science, 2018,27(5):1-7.

[6] W. Yang, X. L. Han. Loading Optimization Problem of Container Liner[J]. Journal of Dalian Maritime University, 2018, 44(1): 72-79.

[7] Z. X. Liu, J. L. Yan, Q. W. Zhang. Optimization of Slot Plan Problem Based on Evolutionary Strategy Algorithm[J]. Computer Engineering and Design. 2019,40(09):2725-2731.

[8] J. Jian, W. J. Mi, M. Y. Xia. Stowage Decision of Container Ships Based on Dual Cycling Process. Journal of Shanghai Maritime University, 2019,40(03):6-13.

[9] J. Li, Y. Zhang, S. Y. Ji, et al. Inland Container Ship Stowage Planning Decision with Multiple Container Types[J]. Journal of Transportation Systems Engineering and Information Technology, 2018, 19(1): 200-207.

[10] W. Tian, Y. Zhang, H. M. Cheng. Research on Sequencing and Bin Packing Problem with Practical Vessel Stowage Constraints[J]. Journal of Wuhan University of Technology (Transportation Science & Engineering), 2016,40(03):509-513).

[11] H. L. Zhu, M. J. Ji, W. W. Guo. Optimization of the Container Retrieving Sequence and Rehandling Strategy Based on Stowage Plan[J]. Journal of Transportation Systems Engineering and Information Technology, 2016, 16(02):191-199).

[12] Bilican M S, Evren R, Karatas M. A Mathematical Model and Two-Stage Heuristic for the Container Stowage Planning Problem With Stability Parameters[J]. IEEE Access, 2020, 8: 113392-113413.

[13] J. L, Y. Zhang, S. Y. Ji, J. M. Dynamic Stowage Planning Decision for Inland Container Liner Route Shipping with Uncertain Container Quantity[J]. Operations Research and Management Science, 2020, 29 (07):64-71.

[14] B. Yang, W. H. Wang, W. T. ZHAO, et al. DQN-based Automatic Stowage Planning Algorithm Using Dynamic Depth Multi-Branch Search[J]. Computer Engineering, 2020, 46 (08): 313-320.

[15] J. Xu, Y. Zhang, J. Li. Research on Full-rout Stowage of Inland Container Vessel based on Granularuty Partition[J]. Journal of Wuhan University of Technology (Transportation Science & Engineering), 2019,43(02):236-241.