Problem Chosen	2021	Team Control Number
A	ShuWei Cup Summary Sheet	202111107503

Summary

With the global spread of the novel coronavirus (COVID-19), ports and shipping industries have been severely hit. The main problem is how to optimize the allocation of port resources and improve the efficiency of port operations as much as possible with reference to the ship load data and berth carrying capacity given in Annex I. In this paper build a heterogeneous ship resource allocation optimization model to solve the problem.

For question 1: Firstly, the expression function of expectation value of all ships based on shipping companies is established from the perspective of shipping companies, and the **heterogeneous ship resource allocation optimization model** with the maximum berth utilization, the maximum expectation value of shipping companies and the maximum fairness is established by considering the capacity of all ships of all shipping companies and the berthing capacity of all berths in one cycle. And solved by **Genetic Algorithm**, the result indicates that the sum of utilization rate of all berths in one cycle is **5489**, and the expectation value of all shipping companies is **3489**.

For question 2: By considering the ship handover time factors of shipping companies at ports, we established a soft **time window-based** heterogeneous ship resource allocation optimization model and designed a time window-based ship overtime penalty strategy. Further, we used the optimization strategy based on **interpolation and aggregation** to solve the problem. The results show that the average completion rate of ship docking plan is **82%** and the vacancy rate of berths is **14.38%**.

For question 3: Based on model 1 and model 2, we established a heterogeneous ship port berth resource allocation model based on docking time which takes the minimum docking time as the main objective. And we used **fast non-dominated ranking strategy** and " **Elimination strategy** " strategy to solve the model.

For question 4: We raised the dimension of this problem to the cooperation between multiple ports in a region, and considered the cooperation between multiple berths of multiple ports and different shipping companies to improve the model we built before. We used **"three-number-neutral" strategy** and "**Elimination strategy**" strategy to solve the model. The result shows that the overall efficiency is maximized when **65%** of each port is available.

For question 5: Based on model of the question 4, we added the port weekend rest constraint. And we also used **"three-number-neutral" strategy** and **"Elimination strategy" strategy** to solve the model. And we produced simulation experiment and verified the model based on in Annex 1. Finally, we presented the results in the “Result presentation” folder.

Keywords:
Heterogeneous ships; Resource allocation optimization; Genetic algorithms; Time windows; Tnterpolation and aggregation strategies; Fast non-dominated sorting strategies; Elimination strategy; Three-number-neutral

Content

1. Introduction

1.1 Background

Marine transportation, as the most important mode of transportation in international trade, plays a vital role in global economic development, and the outbreak and spread of the new coronavirus (COVID-19) has dealt a heavy blow to global economic development and has also profoundly affected the global shipping industry. And with the development of the epidemic prevention and control, the difference of the recovery level of the epidemic between different countries and regions makes a significant difference in the operation of ports and shipping, and in order to balance the utilization situation among berths in each port, alleviate the idle and congested state of ports, and improve the operation level and economic efficiency, the global shipping industry is required to construct a more reasonable strategy for optimizing the allocation of port resources in order to realize the optimization of ports, shipping companies and terminal In order to achieve synergy and mutual benefits among stakeholders such as ports, shipping companies and terminal operators, and to improve the operational efficiency of shipping networks.

1.2 Work

The optimization of port resource allocation involves multiple decision makers such as ports, shipping companies and terminal operators, and to build an optimization strategy that satisfies multiple parties, we need to proceed with joint decision optimization from different perspectives. Therefore, we construct different optimization models for the problem. We also consider the practical constraints in shipping and shipping management process, to analyze and design the optimization strategy rationally. Thus, the problems we are trying to solve are：

- How to provide a solution from the shipping company's point of view: cooperation between shipping companies to relieve port congestion and idleness.
- How to provide an optimal docking strategy for terminal operators and shipping companies from the perspective of terminal operators and design a reward and penalty based on docking time.
- How to provide a ship docking strategy to reduce ship docking time while meeting the working hours of epidemic prevention from the perspective of integrated shipping companies and terminal operators.
- How to link cooperation between multiple ports to provide a solution that reduces port congestion and idleness.
- How to solve the problem again considering multiple ports with 1-2 days off per weekend, and demonstrate with the data in Annex I.

2. Problem analysis

2.1 Data analysis

Annex 1 provides data on 1127 ships' code, carrying capacity, shipping cycle, average docking time, variance of the last ten docking times, the last ten arrival times and berthing capacity of 217 each berth. Through the collation and analysis of the capacity data, we found that the shipping cycle of all ships is distributed from 1 to 14 days, that is, the ships with the shortest cycle arrive at the port every day, while the ships with the longest cycle take 14 days to arrive at the port. And for the data collation and analysis of the last ten arrival times of the ships, we found that the port receives the ships to dock from 8:00 am to 18:00 pm every day, i.e. the ships can only arrive at the port within the specified time.

2.2 Analysis of question 1

The solution to this question requires consideration from the shipping company's point of view, both to meet the needs of all shipping companies and to consider the utilization rate of the berths. The demand of ships varies from one shipping company to another, and since the overall question is to alleviate port congestion and excessive idling before solving the question, how to integrate these factors for optimization is the key to solve this question.

2.3 Analysis of question 2

The solution of this question needs to be fully considered from the perspective of the terminal operator, and the time and berth planning for each ship of the shipping company is carried out under the premise of satisfying the berth utilization rate, so the capacity of different ships of different shipping companies and the carrying capacity of berths are the key factors to be focused on. The key to solving Question 2 is to transform the model perspective of Question 1 into that of a terminal operator, optimize the processing of the time schedule to meet the requirements of Question 2, and finally solve the question.

2.4 Analysis of question 3

The question focuses on the perspective of the need to simultaneously from the perspective of terminal operators and shipping companies, taking into account the premise of epidemic prevention work, while meeting the needs of different shipping companies to reduce the total time of ship docking as much as possible, the strategy provided needs to consider question 1 and question 2 together, and add constraints such as time windows, through the optimization algorithm to solve.

2.5 Analysis of question 4

In question 4, the dimension of the question rises to the cooperation between multiple ports in a region, which is no longer limited to multiple berths in a single port, but needs to consider the cooperative scheduling between multiple berths in multiple ports and different shipping companies, based on the idea of game theory how to optimize the docking time of ships by taking the cooperation between multiple ports as the key, and solving the question by taking the docking time as the objective.

2.6 Analysis of question 5

The consideration of question 5 will continue the actual situation in depth by arranging the port to have 1-2 days of rest for each weekend. Then the model should be improved based on question 4, and the rest time of each port should be planned to meet the basic requirements without delaying the demand of the shipping company's ships, and how to solve this question is the key to this question.

3. Symbol and assumptions

3.1 Symbol description

Parameters	Meanings
p	Port number
i	Shipping company number
j	Ship number
A_g	Berth number
Cij	The j -th ship of shipping company i
CC_{Cij}	The maximum carrying capacity of the ship Cij
	The berthing capacity of the berth A_g
T_{Cij}	The actual time when the ship Cij arrived at the berth
W_p^t	The total benefit values of the port p on the t -th day
Tep	The working time for prevention of disease.

3.2 Fundamental assumptions

1. This paper ignores the type of berths, all of which are of container type. It is known from the literature [1] that the hydraulic structure of the berths is 1.5 times the stated berthing capacity, i.e., the capacity of each berth to serve a single ship shall not exceed 1.5 times the berthing capacity of the berth.
2. To maximize their profits, shipping companies require each ship to perform shipping tasks only when it is fully loaded, i.e., each ship is loaded to its maximum capacity.
3. One ship can only choose one berth.
4. The planned operating cycle of the port is 14 days.
5. We don't plan the ship's track.
6. This paper ignores the ineffective waiting time at berth after loading and unloading of a ship.
7. We get the working hours of the port berth based on Annex I: 8:00 -18:00.

4. Model

4.1 Model of question 1

Decision variables:

$$X_{Cij}^{t} = \begin{cases} 1, \ t\text{-}th\ day\ ship\ Cij\ arrives\ at\ port \\ 0, \ Otherwise \end{cases} \tag{4-1-1}$$

$$Y_{Cij}^{A_g} = \begin{cases} 1, \ Ship\ Cij\ plans\ to\ dock\ at\ berth\ A_g \\ 0, \ Otherwise \end{cases} \tag{4-1-2}$$

$$Z_{(Cij,A_g)}^{h} = \begin{cases} 1, Ship\ Cij\ arrives\ at\ berth\ A_g\ at\ time\ h \\ 0, Otherwise \end{cases} \tag{4-1-3}$$

Objective function:

$$\max\ f_1 = \left(\sum_t \sum_g \frac{\sum_i \sum_j CC_{Cij} \times X_{Cij}^{t} \times Y_{Cij}^{A_g}}{BC_{A_g}}\right) \times \left(\sum_t \sum_i \sum_j E_{Cij}\right) \times \left[\sum_t \sum_i \frac{I}{\sum_j (E_{Cij} - \bar{E})^2}\right] \tag{4-1-4}$$

In Equation (4-1-4), from left to right the first term indicates the sum of the utilization rates of all berths in a planning cycle. The second term indicates the sum of the expected values of all ships of all shipping companies in a planning cycle. The third term indicates the sum of the inverse of the variance of the expected values of all ships of all shipping companies in a planning cycle; CC_{Cij} indicates the maximum carrying capacity of the ship Cij ; BC_{A_g} indicates the berthing capacity of the berth A_g ; E_{Cij}

indicates the expected value of the ship Cij; I indicates the total number of ships of the shipping company; $\bar{E}$ indicates the arrange expected value of the ships.

Where,

$$E_{Cij} = \begin{cases} 1, T_{Ci(j-1)} + TS_{Ci(j-1)} \le T_{Cij} \\ 0, otherwise \end{cases} \tag{4-1-5}$$

$$\beta_{(C_{ij},A_g)}^{(t,h)} = P(h) \times X_{C_{ij}}^{t} \tag{4-1-6}$$

$$P(h) = \frac{1}{h_{\max}} \tag{4-1-7}$$

In Equation (4-1-5), T_{Cij} indicates the actual time when the ship Cij arrived at the berth, $T_{Ci(j-1)}$ indicates the actual time when the ship $Ci(j-1)$ arrived at the berth, $TS_{Ci(j-1)}$ indicates the docking time of the ship $Ci(j-1)$. If the ship $Ci(j-1)$ not stay at the berth when the ship Cij arrived at the berth, the expected value of the ship Cij is 1, otherwise, the expected value of the ship Cij is 0; In Equation (4-1-6), $\beta_{(C_{ij},A_g)}^{(t,h)}$ indicates the probability that the ship Cij will arrive at berth A_g at time h on the t-th day, $P(h)$ indicates the probability that the ship will arrive at each moment.

Constraints:

$$\sum_{t}\sum_{h} \beta_{(Cij,A_g)}^{(t,h)} = N_{Cij} \tag{4-1-8}$$

$$\sum_{j} CC_{Cij} \times X_{Cij}^{t} \times Y_{Cij}^{A_g} \le 1.5BC_{A_g} \tag{4-1-9}$$

$$\sum_{A_g} Y_{Cij}^{A_g} = 1, \ when \ X_{Cij}^{t} = 1 \tag{4-1-10}$$

$$\sum_{h} Z_{(Cij,A_g)}^{h} = 1 \tag{4-1-11}$$

$$\sum_{t}\sum_{g} (1.5BC_{A_g} \times X_{Cij}^{t} \times Y_{Cij}^{A_g}) \ge CC_{Cij} \times N_{C_{ij}} \tag{4-1-12}$$

$$N_{C_{ij}} = \sum_{t} X_{Cij}^{t} \tag{4-1-13}$$

$$8 \le T_{Cij} \le 18, \forall i, j \tag{4-1-14}$$

Constraint (4-1-8) indicates that probability of the ship should satisfy the relationship; Constraint (4-1-9) indicates that the capacity of each berth to serve a single ship cannot exceed 1.5 times the berthing capacity of the berth; Constraint (4-1-10) indicates that each ship can only plan to arrive at a single berth; Constraint (4-1-11) indicates that each ship can only arrive at the berth at one moment; Constraint (4-1-12) indicates the berthing capacity of the berth should satisfy the carrying demand of the ship; Constraint (4-1-13) indicates the number of the ship Cij arrivals at port in a planning cycle; Constraint (4-1-14) indicates that the working hours of the port are from 8:00 to 18:00 and ships can only arrive at the port during this time zone.

4.2 Model of question 2

Objective function:

$$\max f_2 = \left[\sum_t \sum_i \sum_j (\omega_{Cij}^A + \omega_{Cij}^L) \times X_{Cij}^t\right] \times \left(\sum_t \sum_g \frac{\sum_i \sum_j CC_{Cij} \times X_{Cij}^t \times Y_{Cij}^{A_g}}{BC_{A_g}}\right) \times \left[\sum_t \frac{1}{\left(1 - \sum_g X_{Cij}^t \times Y_{Cij}^{A_g} / A_G\right)}\right] \tag{4-2-1}$$

In Equation (4-2-1), from left to right the first term indicates the completion rate of the ship berthing plan in a planning cycle; the second term indicates the summary of the utilization rate of all berths in a planning cycle; the third term indicates the summary of the inverse of the berth vacancy rate in a planning cycle. A_G indicates the total number of berths in the port.

Where,

$$\omega_{Cij}^A = \begin{cases} 1, AT_{Cij} - T_{Cij} = 0 \\ 0, AT_{Cij} - T_{Cij} < 0 \\ -1, AT_{Cij} - T_{Cij} > 0 \end{cases} \tag{4-2-2}$$

$$\omega_{Cij}^L = \begin{cases} 1, LT_{Cij} - (T_{Cij} + TS_{Cij}) \ge 0 \\ -1, LT_{Cij} - (T_{Cij} + TS_{Cij}) < 0 \end{cases} \tag{4-2-3}$$

Where, AT_{Cij} is the s the time when the terminal operator plans for the ship Cij to arrive at the berth; T_{Cij} is the actual time when the Cij arrived at the berth; TS_{Cij} is the docking time of the ship Cij; LT_{Cij} is the time at which the terminal operator plans for the ship Cij to depart. Equation (4-2-2) indicates that the value of ω_{Cij}^A is 1 when the ship arrives at the berth at the scheduled time, the value of ω_{Cij}^A 0 when the ship arrives early, and the value of ω_{Cij}^A -1 when the ship arrives late; Equation (4-2-3) indicates the value of ω_{Cij}^L is 1 when the ship leaves the berth within the planned time and the value of ω_{Cij}^L -1 when the ship leaves the berth later than the planned time.

Constraints:

$$\sum_j CC_{C_{ij}} \times X_{C_{ij}}^t \times Y_{C_{ij}}^{A_g} \le 1.5 BC_{A_g} \tag{4-2-4}$$

$$\sum_{A_g} Y_{C_{ij}}^{A_g} = 1,\ when\ X_{C_{ij}}^t = 1 \tag{4-2-5}$$

$$\sum_h Z^h_{(Cij,A_g)} = 1 \tag{4-2-6}$$

$$\sum_t \sum_g (1.5BC_{A_g} \times X^t_{Cij} \times Y^{A_g}_{Cij}) \ge CC_{Cij} \times N_{C_{ij}} \tag{4-2-7}$$

$$N_{C_{ij}} = \sum_t X^t_{Cij} \tag{4-2-8}$$

$$8 \le T_{Cij} \le 18, \forall i, j \tag{4-2-9}$$

Constraints (4-2-4) - (4-2-9) have the same meaning as constraints (4-1-9) - (4-1-14).

4.3 Model of question 3

Objective function:

$$\min f_3 = \left[\sum_t \sum_i \sum_j (TS_{Cij} \times X^t_{Cij} \times Y^{A_g}_{Cij}) \right] \times \left[\frac{1}{\sum_t \sum_i \sum_j (\omega^A_{Cij} + \omega^L_{Cij}) \times X^t_{Cij}} \right] \tag{4-3-1}$$

In Equation (4-3-1), from left to right, the first term indicates the sum of ship docking time in a planning cycle; the second term indicates the completion rate of ship docking plan in a planning cycle.

$$\alpha_{A_g} = \left(\frac{BC_{A_g} - BC_{A_g} \min}{BC_{A_g} \max - BC_{A_g} \min} \right) \times \alpha_0 \tag{4-3-2}$$

$$TS_{Cij} = \frac{CC_{Cij}}{\alpha_{A_g}} \tag{4-3-3}$$

Where, α_{A_g} is the operational efficiency of the berth A_g ; BC_{A_g} is the berthing capacity for berth A_g , $BC_{A_g} \min$ is minimum of the berthing capacity for berths, and $BC_{A_g} \max$ is maximum of the berthing capacity for berths. $\frac{BC_{A_g} - BC_{A_g} \min}{BC_{A_g} \max - BC_{A_g} \min}$ is the normalization operation; α_0 is the reference efficiency of the berth.

Constraints:

$$\sum_j CC_{C_{ij}} \times X^t_{C_{ij}} \times Y^{A_g}_{C_{ij}} \le 1.5BC_{A_g} \tag{4-3-4}$$

$$\sum_{A_g} Y^{A_g}_{C_{ij}} = 1, \ when\ X^t_{C_{ij}} = 1 \tag{4-3-5}$$

$$\sum_h Z^h_{(Cij,A_g)} = 1 \tag{4-3-6}$$

$$\sum_t \sum_g (1.5BC_{A_g} \times X^t_{Cij} \times Y^{A_g}_{Cij}) \ge CC_{Cij} \times N_{C_{ij}} \tag{4-3-7}$$

$$N_{C_{ij}} = \sum_t X^t_{Cij} \tag{4-3-8}$$

$$8 \le T_{Cij} \le 18, \forall i, j \tag{4-3-9}$$

$$AT_{Cij} - (AT_{Ci(j-1)} + TS_{Ci(j-1)}) \ge Tep \tag{4-3-10}$$

Constraints (4-3-4) - (4-3-9) have the same meaning as constraints (4-1-9) - (4-1-14). Constraint (4-3-10) indicates that the interval between the planned arrival time of ship Cij and the departure time of ship $Ci(j-1)$ should not be less than the working time for prevention of disease; Tep is the working time for prevention of disease.

4.4 Model of question 4

Decision variables:

$$R_{\mathrm{Cij}}^{p} = \begin{cases} 1, \textit{The ship Cij docks at the port p} \\ 0, \textit{Otherwise} \end{cases} \tag{4-4-1}$$

Objective function：

$$\max f_4 = (\sum_t \sum_p W_p^t) \times \left[\sum_t \sum_p (\sum_g \frac{\sum_i \sum_j CC_{Cij} \times R_{(\mathrm{Cij,P})}^{t} \times X_{Cij}^{t} \times Y_{Cij}^{A_g}}{BC_{A_g}}) \right] \tag{4-4-2}$$

In Equation (4-4-2), from left to right, the first term indicates the sum of port benefit values in a planning cycle; the second term indicates the sum of the utilization rates of all berths in a planning cycle.

Where,

$$W_p^t = \sum_i \sum_j (S_{Cij}^{(p,A_g)} \times R_{\mathrm{Cij}}^{p} \times X_{Cij}^{t} \times Y_{Cij}^{A_g}) \tag{4-4-3}$$

$$S_{Cij}^{(p,A_g)} = w_{location} + w_{policy}^{p} + w_{BC}^{(p,A_g)} \tag{4-4-4}$$

$$S_{Cij}^{(p,A_g)} = w_{location} + w_{policy}^{p} + w_{BC}^{(p,A_g)} \tag{4-4-5}$$

$$w_{policy}^{p} = (\frac{BC_{A_g}^{p} - BC_{A_g}^{p} \min}{BC_{A_g}^{p} \max - BC_{A_g}^{p} \min}) \times CC_{Cij} \times \beta_p \tag{4-4-6}$$

$$\beta_p = \beta_0 \times (\frac{CC_{Cij} - CC_{Cij} \min}{CC_{Cij} \max - CC_{Cij} \min}) \tag{4-4-7}$$

Where, W_p^t is the total benefit values of the port p on the t-th day; $S_{Cij}^{(p,A_g)}$ is the benefit value of the port p to select ship; $w_{location}$ is the evaluation index between the ship and the port regarding the location; w_{policy}^{p} is the profitability indicator for port p; $w_{BC}^{(p,A_g)}$ is the berthing indicator of berth A_g of port p; $BC_{A_g}^{p}$ is the berthing capacity of berth A_g of the port p; β_p is the pricing strategy for port p; β_0 is the initial value; $\frac{BC_{A_g}^p - BC_{A_g}^p \min}{BC_{A_g}^p \max - BC_{A_g}^p \min}$ and $\frac{CC_{Cij} - CC_{Cij} \min}{CC_{Cij} \max - CC_{Cij} \min}$ are the normalization operations.

Constraints:

$$\sum_j CC_{C_{ij}} \times R_{\mathrm{Cij}}^p \times X_{\mathrm{Cij}}^t \times Y_{Cij}^{A_g} \le 1.5 BC_{A_g}^p \tag{4-4-8}$$

$$\sum_{A_g} Y_{Cij}^{A_g} = 1,\ when\ R_{\mathrm{Cij}}^p \times X_{Cij}^t = 1 \tag{4-4-9}$$

$$\sum_p R_{(\mathrm{Cij,P})}^t = 1, \forall t \tag{4-4-10}$$

$$\sum_t \sum_p \sum_g (1.5 BC_{A_g} \times R_{\mathrm{Cij}}^p \times X_{Cij}^t \times Y_{Cij}^{A_g}) \ge CC_{Cij} \times N_{C_{ij}} \tag{4-4-11}$$

$$N_{Cij} = \sum_t \sum_p R_{\mathrm{Cij}}^p \times X_{Cij}^t \tag{4-4-12}$$

Constraint (4-4-8) indicates that the capacity of each berth to serve a single ship cannot exceed 1.5 times the berthing capacity of the berth; Constraint (4-4-9) indicates that each ship can only plan to arrive at a single berth; Constraint (4-4-10) indicates that a ship can only complete operations in one port; Constraint (4-4-11) indicates the berthing capacity of the berth should satisfy the carrying demand of the ship; Constraint (4-4-12) indicates the number of the ship Cij arrivals at port in a planning cycle.

4.5 Model of question 5

Assumption:

1. The first day of each planning cycle begins on a Monday.

Decision variables:

$$\mu_p^t = \begin{cases} 1, Port\ p\ opens\ port\ on\ the\ t\text{-}th\ day \\ 0, Otherwise \end{cases} \tag{4-5-1}$$

Objective function:

$$\max f_5 = \left[\sum_t \sum_p (W_p^t \times \mu_p^t) \right] \times \left[\sum_t \sum_p (\sum_g \frac{\sum_t \sum_j CC_{Cij} \times R_{(\mathrm{Cij,P})}^t \times X_{Cij}^t \times Y_{Cij}^{A_g}}{BC_{A_g}} \times \mu_p^t) \right] \tag{4-5-2}$$

Equation (4-5-2) has the same meaning as the equation (4-4-2).

Constraints:

$$\sum_j CC_{C_{ij}} \times \mu_p^t \times R_{\mathrm{Cij}}^p \times X_{\mathrm{Cij}}^t \times Y_{Cij}^{A_g} \le 1.5 BC_{A_g}^p \tag{4-5-3}$$

$$\sum_{A_g} Y_{Cij}^{A_g} = 1,\ when\ \mu_p^t \times R_{\mathrm{Cij}}^p \times X_{Cij}^t = 1 \tag{4-5-4}$$

$$\sum_j \mu_p^t \times R_{(\mathrm{Cij,P})}^t = 1, \forall t \tag{4-5-5}$$

$$\sum_t \sum_p \sum_g (1.5BC_{A_g} \times \mu_p^t \times R_{\mathrm{Cij}}^p \times X_{Cij}^t \times Y_{Cij}^{A_g}) \geq CC_{Cij} \times N_{C_{ij}} \tag{4-5-6}$$

$$N_{Cij} = \sum_t \sum_p \mu_p^t \times R_{\mathrm{Cij}}^p \times X_{Cij}^t \tag{4-5-7}$$

$$\sum_p \mu_p^t = 1, t \in \{1, 2, \cdots, 5, 8, 9, \cdots, 12\} \tag{4-5-8}$$

$$5 \leq \sum_t \mu_p^t \leq 6 \tag{4-5-9}$$

$$\sum \mu_p^t > 0, t \in \{6, 7, 13, 14\} \tag{4-5-10}$$

Constraints (4-5-3) - (4-5-7) have the same meaning as constraint (4-4-8) - (4-4-12); Constraints (4-5-8) indicate that Ports cannot be closed Monday through Friday; Constraint (4-5-9) indicates that each port can only rest for 1-2 days; Constraint (4-5-10) indicates that the port can be closed on Saturdays or Sundays.

5. Test the models

5.1 Test of the model of question 1

5.1.1 Solution ideas

The arrival time and berth selection of each ship are decided by each shipping company, and the shipping company only knows the berthing capacity of each berth in advance so that it can choose by itself, but it does not know the berth selection of other shipping companies, and each ship cannot know the arrival time of other ships. Preprocessing data, the matrix of berths that can be reached by each ship is processed with equal probability, and a random series is obtained using the randsrc function and randomly assigned. The port operation time from 8:00 to 18:00 is divided into 10 equal parts, and the probability of each ship's arrival time is specified to be the same, and subsequently a random bit sequence is generated for berth selection. The ship arrival expectation, berth utilization, and the inverse of the expected variance are recorded and calculated. The berth selection is used as the independent variable and finally the optimization process is performed using genetic algorithm to solve for the maximum value of the objective function. Figure 1 indicates the solving algorithm flow of the problem one.

5.1.2 Solving process

Step 1：Define the data matrix；

Step 2：Pre-process data, correct data matrix features, retain key data such as row counts；

Step 3：Using a 14-day cycle, 30 companies, 40 ships per company as the base data, determine the number of ships arriving each day in the cycle and their numbers；

Step 4：Assign arrival random time slots and create an empty table to record the sum of arrival time and stopping time；

Step 5：Repeat detection of berth selection to determine whether a later arriving ship needs to wait for a first arriving ship to leave. Create an all-one matrix, and record the expected value of the ship that needs to wait or transfer as 0；

Step 6：Calculate the sum of the berth utilization, the sum of expected value, and the sum of the expected value inverse of the variance；

Step 7：The berth selection matrix is used as the initial base data for optimization. Use the product of berth utilization, desired sum, and inverse sum of desired variances as the objective function and use a genetic algorithm for optimization.

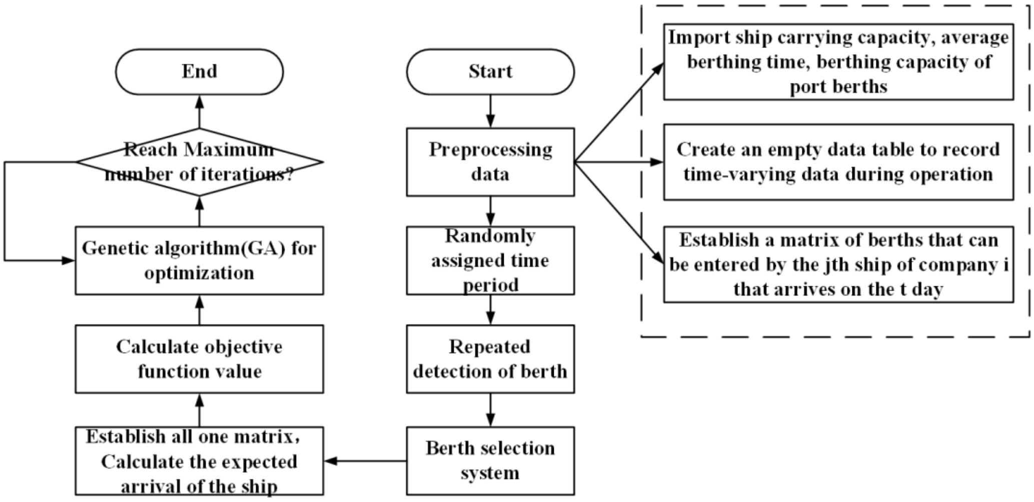

Figure 1 The solving algorithm flow of the problem one

5.1.3 Solution results

Table 1 indicates the worst, optimal, and average objective function values for 50 runs. Figure 2 shows the worst-case iteration graph for 50 runs. Figure 3 shows the optimal-case iteration graph for 50 runs. Figure 4 shows the ascending order of optimization results for 50 runs.

Table 1

The worst, optimal, and average objective function values for 50 runs.

	Target expectations	Total berth utilization	Objective function value ($\times10^{13}$)
Worst	3403	4694.492	5.996
Optimal	3498	5489.347	16.25
Average	3470	5247.887	9.409

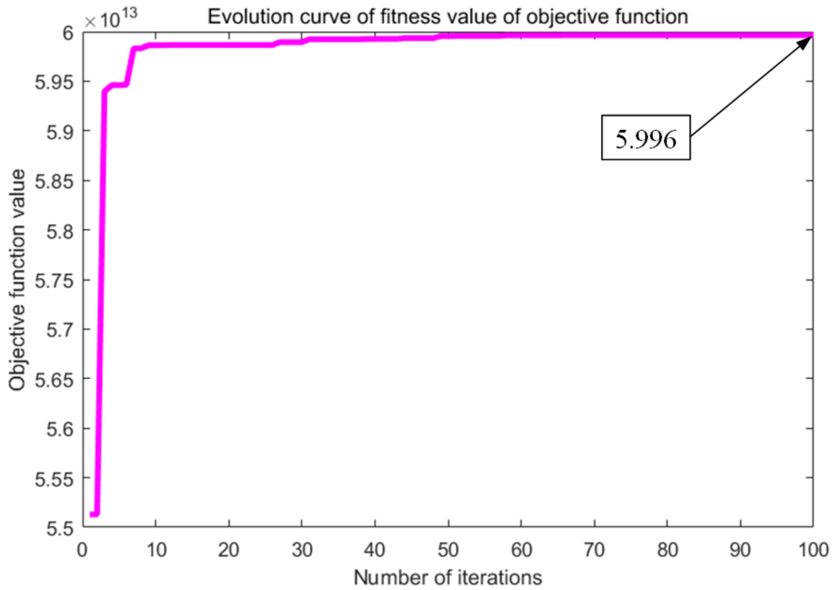

Figure 2 The worst-case iteration graph for 50 runs

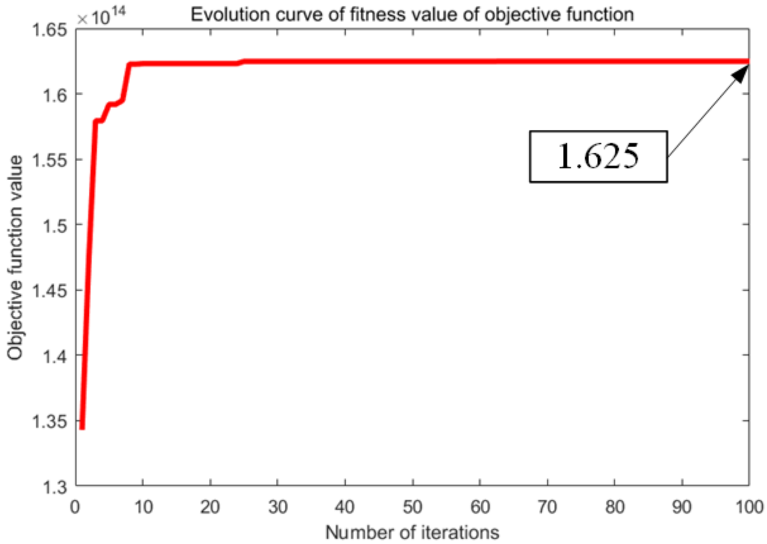

Figure 3 The optimal-case iteration graph for 50 runs.

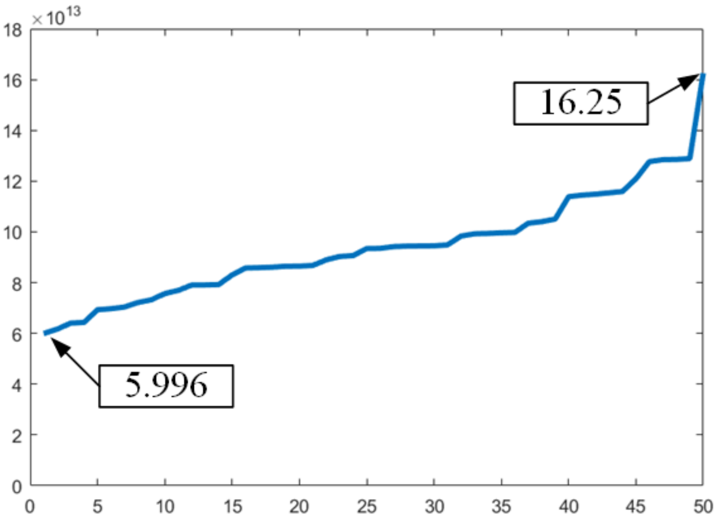

Figure 4 The ascending order of optimization results for 50 runs

5.2 Test of the model of question 2

5.2.1 Solution ideas

Question 2 shifts the focus to the port, adding to the first question the autonomous port time planning, where the port gives a scheduled arrival time and a scheduled departure time to the ships arriving each day. Similarly, by each ship being able to select one of them with equal probability according to the berths it can enter, considering the repeated selection of ships from different companies, the port operation time 8:00-18:00, etc. is divided into ten time periods, and the arrival time is randomly assigned with equal probability, so that the actual arrival time of the ship as well as the departure time can be known. Further, calculate the sum of the completion rate of ship docking plan, the sum of the utilization rate of all berths, and the sum of the inverse of the berth vacancy rate and use the product of the three factors above as the objective function. Return the optimized value matrix to the next cycle as the initial value using the optimization strategy of interpolation and aggregation.

5.2.2 Solving process

Step 1： Preprocess the data and initialize the planned arrival time given by the port;

Step 2： Perform step 1-4 in the same question 1, and the berth selection and repeat detection;

Step 3： Enter the balance constraint of the ship at the berth;

Step 4： Calculate the initial objective function value, and return to the planned time given by the port after a cycle of calculation;

Step 5： Optimize the selection of berths, divide the utilization range into 3 layers, and select according to the priority from high to low;

Step 6： Optimize the planning time, get the returned planning time value from step 4, and calculate the expectation of completion rate, thereby giving a new planning time.

Step 7： Repeat the optimization process to achieve the best.

Figure 5 shows the flow chart for solving question two. Figure 6 the berth state balance process. Figure 7 shows the expectation optimization strategy.

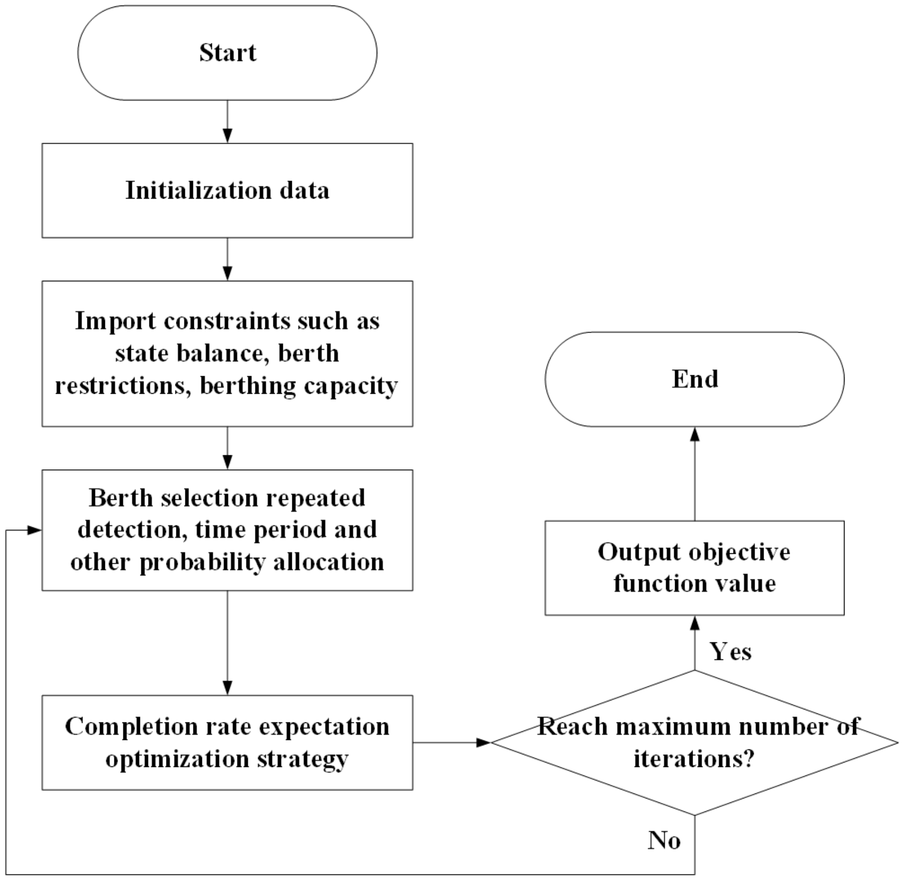

Figure 5 The flow chart for solving question two

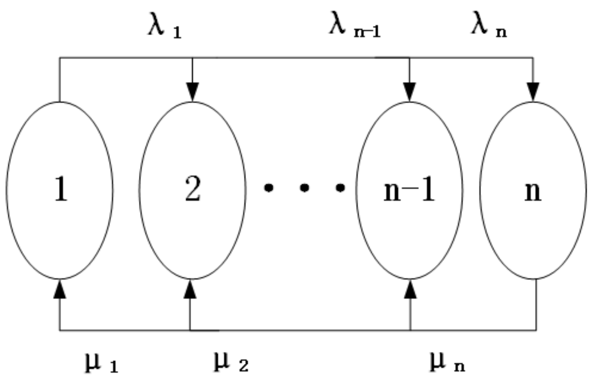

Figure 6 The berth state balance process

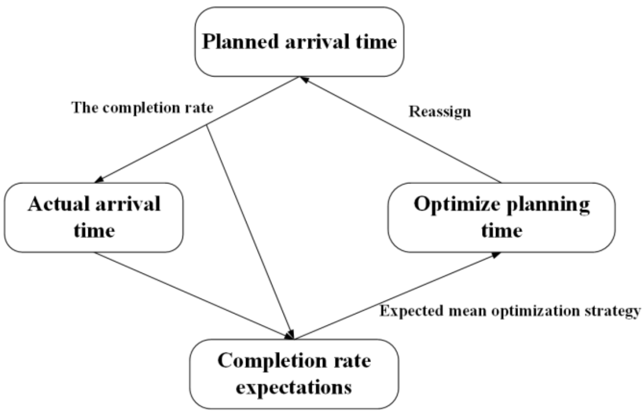

Figure 7 The expectation optimization strategy diagram

5.2.3 Solution result

Figure 8 is a visual display of the optimal completion rate of 50 runs. Figure 9 shows the best result curve of 50 runs. Figure 10 is a visual display of the reciprocal of the berth vacancy rate in one cycle in the optimal results of 50 runs.

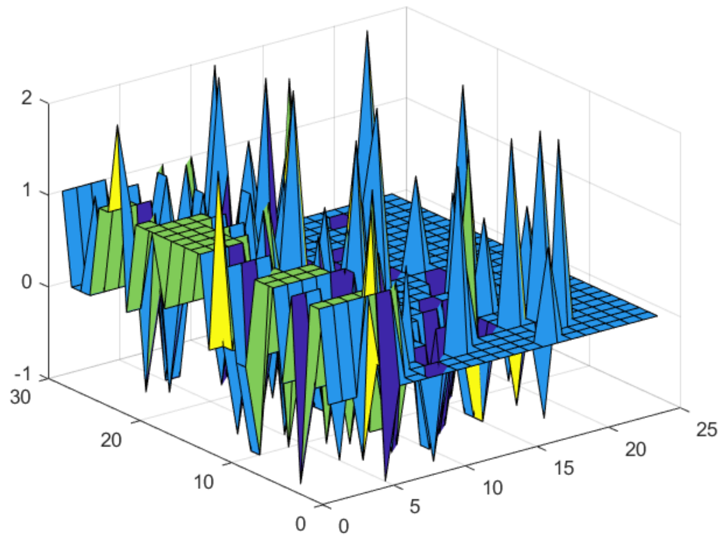

Figure 8 The visual display of the optimal completion rate of 50 runs.

Table 2

Optimal running result data

Ship docking plan completion rate	Sum of berth utilization	Sum of reciprocal berth vacancy rate	Objective function value
82	$4.1919*10^3$	31.045	1.066*107

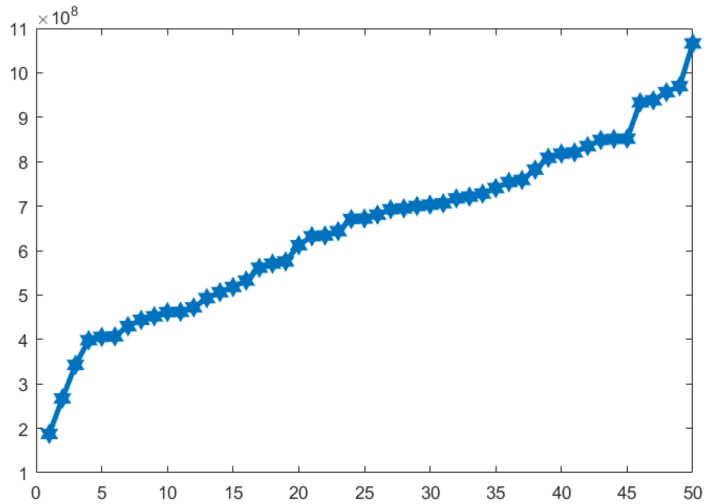

Figure 9 The best result curve of 50 runs

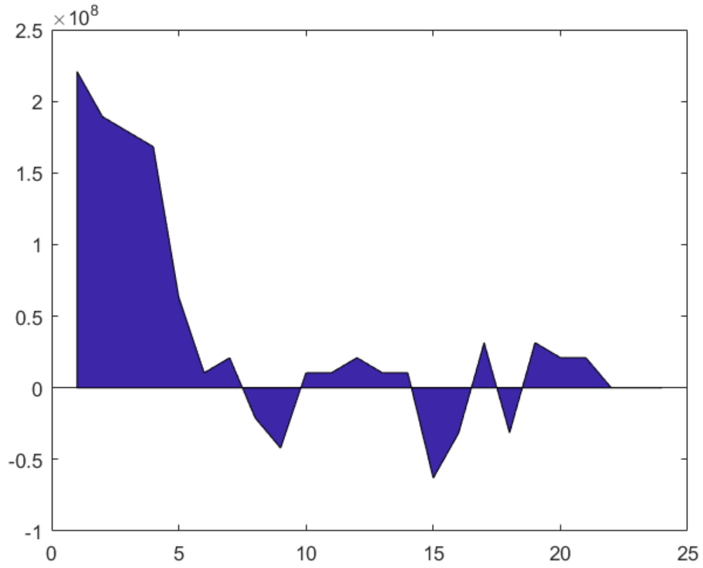

Figure 10 A visual display of the reciprocal of the berth vacancy rate in one cycle in the optimal results of 50 runs

Through the analysis of the above results, we can find that from the perspective of the terminal operator, after the reward and punishment mechanism is proposed, the impact of the operational planning strategy is more obvious. The value of each score (completion rate) can clearly show the optimization process and result optimization situation. From the time docking arrangement, after the planned arrival and departure time are given by the port, the actual arrival time of the ship usually does not match the planned arrival, but the departure time is mostly close to the planned departure time. This shows that the docking between the ship and the port is very important, and the port needs to set the arrival time with the ship in advance. In addition, after repeatedly checking the data results, we propose key improvements to improve the efficiency of the docking according to the optimization degree of the strategy:

From the optimization degree situation：

1) Using the previous optimization result as the initial value in the optimization process can significantly improve the result of the next optimization.

2) The incentive and penalty mechanisms are clearly reflected, visually reflecting the importance of ship-port docking.

In terms of actual efficiency improvements:

1) The port needs to contact the shipping company in advance, so that the planned arrival time can be arranged to achieve the highest utilization rate of the port berth.

2) Shipping companies need to analyze the berthing capacity of ports in advance and make decisions in advance.

3) Ports need to disclose berthing options and arrival times among shipping companies.

5.3 Test of the model of question 3

5.3.1 Solution ideas

Problem 3 aims to optimize the ship docking strategy from three perspectives: port berths, terminal operators and shipping companies, with the objective function of the ship docking time in the port. The average docking time of a ship has been given in the table in Annex I, but this docking time does not reflect the waiting time of a ship at the same berth and the assignment time in the port. So, we get new index values as docking time for each ship based on the ship's capacity and berthing capacity of the berth. The inverse of the completion rate of the ship call schedule is used as the relevant numerical matrix, and a fast-non-dominated ranking and a “Elimination strategy” are added to Problem 2.

5.3.2 Solving process

Step 1：Preprocess the data, sort and statistically process the relevant data in Annex 1.

Step 2：Perform step 1-4 in the process of solving problem one. When initializing the berth selection, the optimization strategy in problem 2 is directly introduced, and the initial optimization result is from the first generation.

Step 3：Introduction of epidemic prevention work time, limiting the minimum time in the ship's docking time to achieve epidemic prevention work constraint.

Step 4：The sum of ship docking time and the inverse sum of ship docking schedule completion rate are calculated, and the product of the two is used as the objective function for cyclic optimization, and the maximum number of iterations is set to 100.

Step 5：Output objective function value.

5.3.3 Solution result

Table 3 shows the worst, best and average results data in the 100 runs. Figure 11 shows the best results for 100 runs.

Table 3

The worst, optimal and average results data in the 50 runs.

	Objective function value	Difference from average
Worst	6344	210
Optimal	5920	-214
Average	6134	Inf

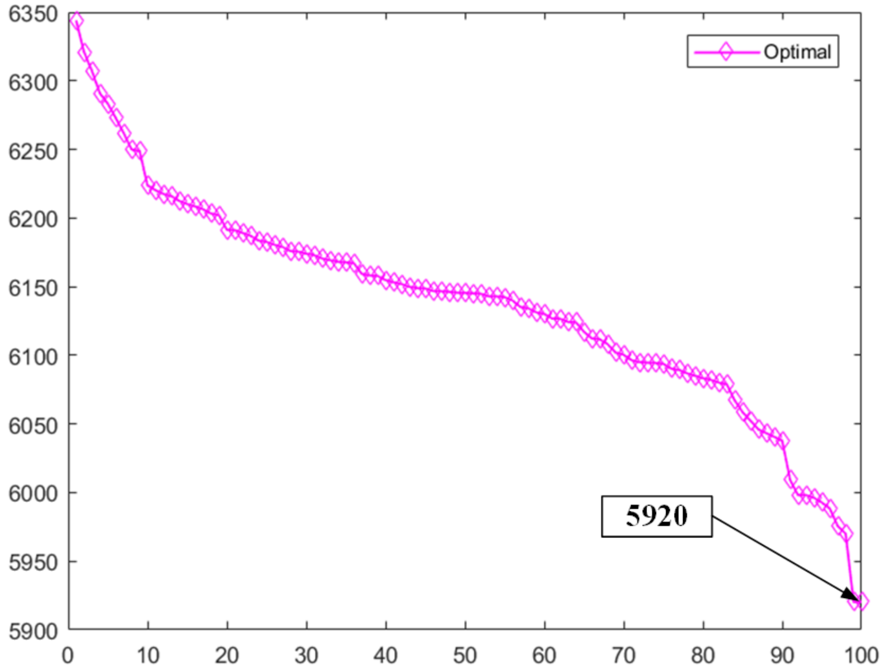

Figure 11 The best results for 100 runs

From Figure 11, we can find that the cycle ends when the optimal result reaches the 96th generation. By repeatedly bringing in the data, it was found that at the 96th generation, the ship docking schedule completion rate did not reach 45% of the satisfied condition. This is only higher than the initial value of 27%, at which point the evolution rate becomes zero, and the iterative optimization process is stopped. We then used the initial data from the optimal run results after giving the optimization from the 2nd generation as the starting value, i.e., excluding the values from the first generation, and calculated the average value during the optimal run, as well as the optimization. The results show that without reducing the prevention time (0.5h), the prevention time has much less influence on the results than other influencing factors.

5.4 Test of the model of question 4

5.4.1 Solution ideas

In calculating the total benefits, since no evaluation index is given, we give the size of the evaluation index for each berth in each port according to the assignment criterion of random probability. Pre-experiments were first conducted to seek a range of values that wanted to be stable and meaningful for implementation by constantly changing the range of values of evaluation indicators, which was finally set to [1, 5]. When setting the benchmark selection system framework, the basic system selection and other situations are the same as the third question. In processing the data, after normalizing the berthing capacity of the berths, the profitability index of each port is then set. Using the same way, based on the berthing capacity of berths in Annex I, the minimum value of berthing capacity is found to be 100 and the maximum value is 600000, and after calculating the mean and median of the berthing capacity array, it is divided into five intervals, and the berthing index is taken as [1,5] after arranging the interval values from small arrivals. Finally, due to the different initial values of the optimization of the total benefit value and the utilization rate, different optimization methods are chosen to optimize both in blocks.

5.4.2 Solving process

Step 1：Statistical partitioning process to statistically partition the berthing capacity of a berth and give berthing indexes according to the mean and median values.

Step 2：The solution steps are the same as step 1-4 of problem 2.

Step 3：The total benefit value of the port is calculated based on Equation (4-4-3), and the sum of the utilization rates of all berths in the port is calculated based on Equation (4-4-2).

Step 4：Optimize the matrix data in chunks based on the " three-number-neutral" array matrix optimization strategy, and perform the " Elimination strategy" selection.

Step 5：Output optimization results.

5.4.3 Solution results

Table 4 shows the detailed simulation results of the best results in 50 times. Figure 12 shows the change value of a utilization optimization result. Figure 13 shows the benefit distribution of each berth in each port, where the left graph reflects the specific gain per berth in each port, and the size of the area in each division in the right graph represents the degree of gain in each port (the sum of the evaluation indicators of the incoming ships), with the degree of intensity representing the size of the gain.

Table 4

Detailed simulation results of the best results in 50 times

	Utilization rate	Total benefit	Objective function value
Numerical value	$5.326*10^3$	$5.902*10^7$	$3.143*10^{11}$
Percentage of benefit improvement after a single port berth exceeds 65% (reserve five decimal places)	0.01013	0.04837	0.03796

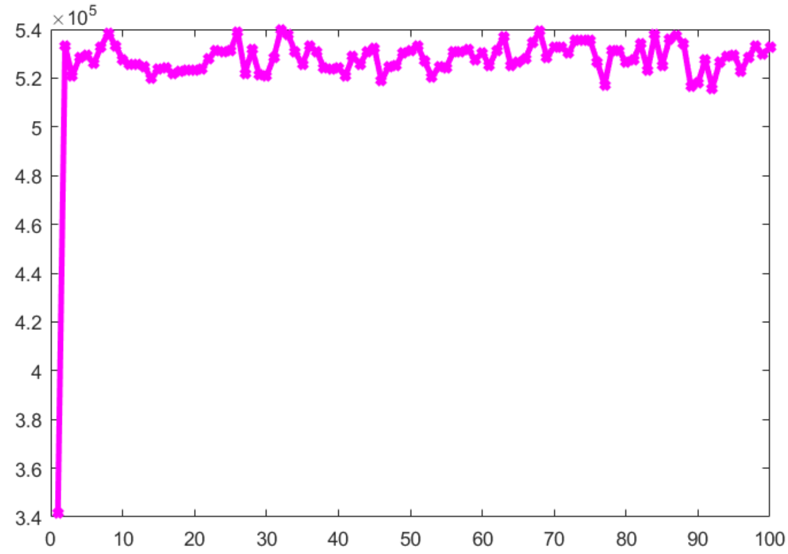

Figure 12 The change value of a utilization optimization result

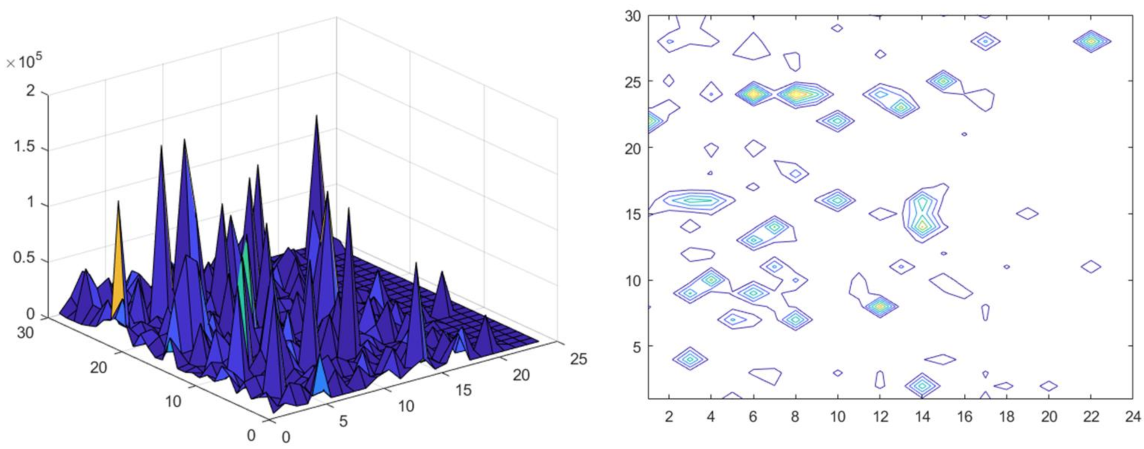

Figure 13 The benefit distribution of each berth in each port

By analyzing the above results, we find that the factor with more influence on the results is the sum of benefits. This indicates that the positive benefits brought by port cooperation within a region are greater than the benefits of reasonable distribution between individual port berths and ships. From a realistic point of view, the simulation results have a strong mapping significance. In a city-port cooperation, the docking time of a ship entering a port is influenced by many factors, and when the berth idle rate of a port is reduced to a certain value, the port will become "crowded". Therefore, we suggest that when 65% of a port's berths are available, it is more beneficial for the port to direct ships to other ports.

5.5 Test of the model of question 5

5.5.1 Solution idea

Question 5 adds the constraint that some ports are closed for 1-2 days on weekends to Question 4, making the number of ports available for selection less. Therefore, only this constraint needs to be added to the fourth question. The basic strategy does not need to be changed.

5.5.2 Solving the results

Since the results vary from run to run, we ran the program 50 times and selected the result of one of the targets optimal as the overall effect. We saved three data in the optimal one run: the number of ships arriving each day, the number of ports selected (20 ports), and the berths selected in that port, where the number of rows represents the number of companies and the columns represent the data of each content. Due to the large amount of data, we only show the data for days 1, 4, 9 and 14, and the complete data has been saved in separate packages. Table 5 shows Port docking optimization model data display. Table 6 shows the partial display of optimal results among 50 times. Figure 14 shows the run 100 times and the smoothed curve.

Note: Question 5 has been saved to 'Result presentation' folder.

Tab 5

Port docking optimization model data display

Project	Utilization rate	Total benefits	Objective function value
Numerical value	$5.173*10^3$	$6.092*10^7$	$3.151*10^{11}$

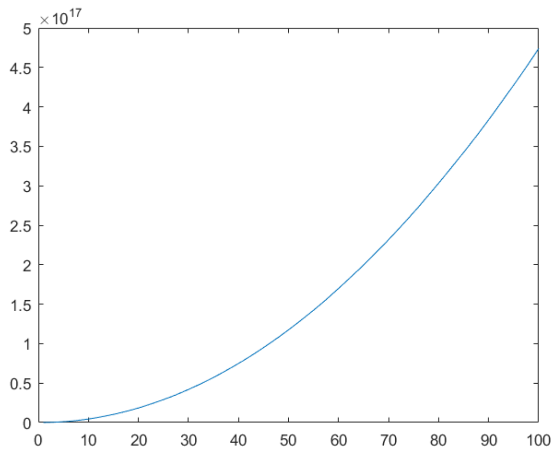

Figure 14 Run 100 times and the smoothed curve

Tab 6

Partial display of optimal results among 50 times

	Shipping companies	Ship number	Port Number	Berth
Day1	1	ZH31	19	BW140
		ZH481	19	BW45
		•••		
	30	ZH540	6	BW110
		ZH870	3	BW14
•••		•••		
Day4	1	ZH31	4	BW175

			•••	
		ZH841	19	BW157
		•••		
		ZH150	7	BW71
	30		•••	
		ZH1170	3	BW111
•••		•••		
		ZH31	3	BW76
	1		•••	
		ZH1021	10	BW192
Day9		•••		
		ZH180	9	BW57
	30		•••	
		ZH1140	2	BW213
•••		•••		
		ZH31	2	BW96
	1		•••	
		ZH1050	15	BW75
Day14		•••		
		ZH60	9	BW216
	30		•••	
		ZH1170	3	BW162

6. Strengths and weakness

6.1 Strengths

1. This paper establishes a resource allocation optimization model from the perspectives of shipping companies, terminal operators and port berths, respectively, in response to the problem requirements, considering the actual shipping situations such as ship cyclical operations and port berth heterogeneity, and derives a more generalized model with strong practical reference value.

2. This paper constructs multi-dimensional optimization indexes such as berth utilization rate, ship expectation value, plan completion rate and port benefit value, which makes the established resource allocation optimization model more comprehensive and effective.

6.2 Weakness

1. The paper ignores the influence of port layout on ship berthing and idealizes the type of ship, actual loading and berth type, which reduces the practicality.

2. This paper does not consider the chain effects on subsequent ships due to the failure of ships to complete their operations within the planned time window, and the accuracy needs to be improved due to the lack of actual operation data of ships and port berths.

7. Conclusion

In this paper, we focus on the optimization of port resource allocation under the influence of novel coronavirus. We first summarized and analyzed the data given in the annex, and obtained the ship load characteristics and berth berthing characteristics, and analyzed the periodic operation of ships. Based on the above analysis results, we focus on the requirements of the competition from the perspective of decision makers such as shipping companies and terminal operators respectively, and consider the actual operational constraints such as port berth heterogeneity, ship heterogeneity and anti-epidemic working time. By constructing the optimization indexes such as berth utilization rate, ship expectation value and plan completion rate, we established the port resource allocation optimization models respectively, and completed the solution of the models using genetic algorithm, optimization strategies based on interpolation and aggregation, and fast non-dominated sorting strategies respectively. It effectively reduces the congestion of the port and improves the utilization efficiency of the port berths. Further, we construct a port efficiency function based on the idea of cooperative game in game theory, and propose a cooperative rest strategy to generalize the established model and obtain a cooperative resource allocation optimization model among multiple ports. The model is solved by using the optimization strategy based on Random quick sort "three-number-neutral" operation and "superiority-worst" operation. The practicality and effectiveness of the model are verified, and it is an effective reference and reference value for solving the port resource allocation optimization problem in the context of the new crown pneumonia epidemic.

References

[1] Zeng Huazhang, Cui Guochun, Xu Yufu. Analysis of berthing capacity of terminals in maritime supervision[J]. China Maritime,2020(11):67-69.

[2] Jiang Qiyuan, Xie Jinsing, Ye Jun, Mathematical modeling [M]. Beijing: Higher Education Press, 2018.5.

[3] Yu Shaoquan, Yang Diwei, Experimental foundations of mathematical modeling [M]. Wuhan: China University of Geosciences Press, 2019.7.

[4] Si Shoukui, Sun Zhaoliang, Mathematical modeling algorithms and applications. [M] Beijing: National Defense Industry Press, 2020.2.

[5] Li Xin, MATLAB mathematical modeling. [M] Beijing: Tsinghua University Press, 2017.

Appendix

Note:Procedure Standard_value.m and repeat.m are used in Questions 1 to 5.
Question1
main.m

```
clear all;
close all;
clc;
A=[];
B=[];
C=[];
for gg=1:50
run question1
z=30;
c=184;
Xs1=rand*(3503-3450)+3450;
Xx1=50;
NP=z;
L=c;
Pc=0.8;
Pm=0.1;
G=100;
f=randi([0,1],NP,L)
for k=1:G
    %Decode binary to decimal within the domain
    for i=1:NP
        U=f(i,:);
        m=0;
        for j=1:L
            m=U(j)*2^(j-1)+m;
        end
        x(i)=Xx1+m*(Xs1-Xx1)/(2^L-1);
        Fit(i)= fitness1(x(i));
    end
    maxFit=max(Fit);
    minFit=min(Fit);
    rr=find(Fit==maxFit);
    fBest=f(rr(1,1),:);
    xBest=x(rr(1,1));
    Fit=(Fit-minFit)/(maxFit-minFit);
    sum_Fit=sum(Fit);
    fitvalue=Fit./sum_Fit;
    fitvalue=cumsum(fitvalue);
    ms=sort(rand(NP,1));
    fiti=1;
    newi=1;
    while newi<=NP
        if (ms(newi))<fitvalue(fiti)
            nf(newi,:)=f(fiti,:);
            newi=newi+1;
```

```
        else
            fiti=fiti+1;
        end
    end
    for i=1:2:NP
        p=rand;
        if p<Pc
            q=randi([0,1],1,L);
            for j=1:L
                if q(j)==1;
                    temp=nf(i+1,j);
                    nf(i+1,j)=nf(i,j);
                    nf(i,j)=temp;
                end
            end
        end
    end
    i=1;
    while i<=round(NP*Pm)
        h=randi([1,NP],1,1);
        for j=1:round(L*Pm)
            g=randi([1,L],1,1);
            nf(h,g)=~nf(h,g);
        end
        i=i+1;
    end
    f=nf;
    f(1,:)=fBest;
    trace1(k)=maxFit;
end
%Second optimization
xBest;
z=30;
c=184;
Xs1=1.50*(rand*(3503-3450)+3450);
Xx1=50*0.5;
NP=z;
L=c;
Pc=0.8;
Pm=0.1;
G=100;
f=randi([0,1],NP,L)
for kk=1:G
    for i=1:NP
        U=f(i,:);
        m=0;
        for j=1:L
            m=U(j)*2^(j-1)+m;
        end
        x(i)=Xx1+m*(Xs1-Xx1)/(2^L-1);
```

```
            Fit(i)= fitness1(x(i));
        end
        maxFit=max(Fit);
        minFit=min(Fit);
        rr=find(Fit==maxFit);
        fBest=f(rr(1,1),:);
        xBest=x(rr(1,1));
        Fit=(Fit-minFit)/(maxFit-minFit);
        sum_Fit=sum(Fit);
        fitvalue=Fit./sum_Fit;
        fitvalue=cumsum(fitvalue);
        ms=sort(rand(NP,1));
        fiti=1;
        newi=1;
        while newi<=NP
            if (ms(newi))<fitvalue(fiti)
                nf(newi,:)=f(fiti,:);
                newi=newi+1;
            else
                fiti=fiti+1;
            end
        end
        for i=1:2:NP
            p=rand;
            if p<Pc
                q=randi([0,1],1,L);
                for j=1:L
                    if q(j)==1;
                        temp=nf(i+1,j);
                        nf(i+1,j)=nf(i,j);
                        nf(i,j)=temp;
                    end
                end
            end
        end
        i=1;
        while i<=round(NP*Pm)
            h=randi([1,NP],1,1);
            for j=1:round(L*Pm)
                g=randi([1,L],1,1);
                nf(h,g)=~nf(h,g);
            end
            i=i+1;
        end
        f=nf;
        f(1,:)=fBest;
        trace2(kk)=maxFit;
end
xBest;
for h=1:100
```

```
trace(h)=trace1(h)*trace2(h)*variance_sum;
end
figure
plot(trace)
xlabel('Number of iterations')
ylabel('Objective function value')
title('Evolution curve of fitness value of objective function')
A=[A,trace(100)];
% saveas(gcf,['Code\Question1\','fig',num2str(gg)])
end
B=sort(A);
figure
plot(B)
[~,a]=find(A == min(A))
[~,b]=find(A == max(A))
c=sum(A)/50;
plot(B)
```

solve1.m

```
load ZH
load ZH_Ori
load BW
load Date
load ZH_wait
%% Data processing before start
pi=[];
for l=1:10
	pi=[pi,0.1];
end
[BW_row,~]=size(BW);
%% Determine the arrival ship number on the day
m=[];
day_arrive=[];
N_sum_not0=[];
N_m_row=[];
EEE=[];
E_sum_cycle=[];
select_Uti_cycle=[];
N_EEE_row=[];
N_EEE_col=[];
SELECT=[];
for xhy=1:14
	%% Determine which ships will arrive currently
	select=[];
	day=xhy;
	E1=[];
	EEE_row=[];
	EEE_col=[];
	m=[];
	ZH_day=[];
```

```
day_arrive=[];
sum_not0=[];
for ijj=1:14
    dayy=rem(day,ijj);
    if dayy == 0
        day_arrive=[day_arrive,ijj];
    end
end
[~,day_arrive_row]=size(day_arrive);
for q=1:day_arrive_row
    m=[m;find(Date==day_arrive(q))];
end
m=sort(m);
[m_row,~]=size(m);
N_m_row=[N_m_row,m_row];
for ii=1:30
    num=1;
    for jj=1:m_row
        cycle=0;
        for ij=1:40
            if m(jj) == (ii+cycle)
                ZH_day(ii,num)=m(jj);
                num=num+1;
            end
            cycle=cycle+30;
        end
    end
end
day_random=[];
for ii=1:30
    sum_not0(ii)=sum(sum(ZH_day(ii,:)~=0));
    %% Time of arrival probability
    day_random(ii,:)=randsrc(1,max(30),[8:17;pi]);
end
N_sum_not0=[N_sum_not0;sum_not0];
%% Each ship chooses a port berth
det_berth=[];
for i=1:30
    if sum_not0(i) == 0;
        continue
    end
    for n=1:sum_not0(i)
        det_berth=[];
        for j=1:BW_row
            if ZH_Ori(ZH_day(i,n)) < (1.5*BW(j))
                det_berth=[det_berth,j];
            end
        end
        [~,det_berth_row]=size(det_berth);
        pii=[];
```

```
                for jj=1:det_berth_row
                    pii=[pii,1/det_berth_row];
                end
                BW_random=randsrc(1,1,[1:det_berth_row;pii]);
                if n == 1
                    select(i,1)=det_berth(BW_random);
                end
                if n > 1
                    for cycle1=1:n
                        [answer,select]=repeat(select,i,n,det_berth,det_berth_row,pii);
                        if answer == 1
                            break
                        end
                    end
                end
                %% Calculate ship utilization
                select_Uti(i,n)=ZH_Ori(ZH_day(i,n))/BW(select(i,n))*100;
            end
        end
        SELECT=[SELECT,select]
        %% Calculate the expected arrival of each ship
        select_2=select;
        select_2(find(select_2==0))=[];
        select_same=tabulate(select_2);
        select_same_num=find(select_same(:,2) > 1);
        [select_same_row,~]=size(select_same_num);
        time_num=[];
        for g=1:select_same_row
            [select_i,select_j]=find(select == select_same_num(g));
            [select_i_row,~]=size(select_i);
            E=select;
            E(find(E ~= 0))=1 ;
            for h=1:select_i_row
                time(h)=day_random(select_i(h),select_j(h));
                time_num=[time_num;time(h),select_i(h),select_j(h)];
            end
            first_time=min(time);
            first=find(time_num(:,1) == first_time);
            for y=1:select_i_row
                if y ~= first
                    E(select_i(y),select_j(y))=0;
                end
            end
        end
        %% Calculate the total expectations of all shipping companies on the day
        E_sum=sum(sum(E));
        E1=E;
        EEE=[EEE,E1];
        [EEE_row,EEE_col]=size(EEE);
        N_EEE_col=[N_EEE_col,EEE_col];
```

```
        %% Calculate the total utilization of all ships on the day
        select_Uti_sum=sum(sum(select_Uti)); %当天总利用率
        %% Calculate all data for the day
        E_sum_cycle=[E_sum_cycle;E_sum];
        select_Uti_cycle=[select_Uti_cycle;select_Uti_sum]
end
%% Calculate the sum of the reciprocal variances of the expected values of all ships
in a period
E_average=sum(E_sum_cycle)/sum(N_m_row);
for jjj=1:14
        num2=1;
        for b=1:N_EEE_col(jjj)
                variance(jjj,b)=1/((EEE(jjj,num2)-E_average)^2);
                num2=num2+1;
        end
end
variance_sum=sum(sum(variance));
E_sum_cycle1=sum(sum(EEE));
select_Uti_cycle1=sum(select_Uti_cycle);
```

fitness1.m

```
function result=fitness1(x)
fit= sum(x);
result=fit;
end
```

fitness2.m

```
function result=fitness2(x)
fit= sum(x);
result=fit;
End
```

Question2

solve2.m

```
clear all
close all
clc
N_trace=[];
%%
load ZH
load ZH_Ori
load BW
load Date
load ZH_wait
%%
pi=[];
for l=1:10
        pi=[pi,0.1];
end
[BW_row,~]=size(BW);
%%
```

```
m=[];
day_arrive=[];
N_sum_not0=[];
N_m_row=[];
EEE=[];
E_sum_cycle=[];
select_Uti_cycle=[];
N_EEE_row=[];
N_EEE_col=[];
SELECT=[];
N_Berth_vacancy=[];
for xhy=1:14
    %%
    select=[];
    day=xhy;
    E1=[];
    EEE_row=[];
    EEE_col=[];
    m=[];
    ZH_day=[];
    day_arrive=[];
    sum_not0=[];
    for ijj=1:14
        dayy=rem(day,ijj);
        if dayy == 0
            day_arrive=[day_arrive,ijj];
        end
    end
    [~,day_arrive_row]=size(day_arrive);
    for q=1:day_arrive_row
        m=[m;find(Date==day_arrive(q))];
    end
    m=sort(m);
    [m_row,~]=size(m);
    N_m_row=[N_m_row,m_row];
    for ii=1:30
        num=1;
        for jj=1:m_row
            cycle=0;
            for ij=1:40
                if m(jj) == (ii+cycle)
                    ZH_day(ii,num)=m(jj);
                    num=num+1;
                end
                cycle=cycle+30;
            end
        end
    end
    day_random=[];
    for ii=1:30
```

```
          sum_not0(ii)=sum(sum(ZH_day(ii,:)~=0));
          %%
          day_random(ii,:)=randsrc(1,max(30),[8:17;pi]);
     end
     N_sum_not0=[N_sum_not0;sum_not0];
     %%
     det_berth=[];
     for i=1:30
          if sum_not0(i) == 0;
               continue
          end
          for n=1:sum_not0(i)
               det_berth=[];
               for j=1:BW_row
                    if ZH_Ori(ZH_day(i,n)) < (1.5*BW(j))
                         det_berth=[det_berth,j];
                    end
               end
               [~,det_berth_row]=size(det_berth);
               pii=[];
               for jj=1:det_berth_row
                    pii=[pii,1/det_berth_row];
               end
               BW_random=randsrc(1,1,[1:det_berth_row;pii]);
               if n == 1
                    select(i,1)=det_berth(BW_random);
               end
               det_day_random(i,n)=day_random(i,n);

[scole_CA,scole_CL]=Standard_value(det_day_random,n,i,ZH_day,ZH_wait);
               scole_end(i,n)=scole_CA(i,n)+scole_CL(i,n);
               if sum(scole_end(i,:)) == 0
                    for ff=1:ceil(n/2)
                         scole_end(i,ff)=1
                    end
               end
               if n > 1
                    for cycle1=1:n

[answer,select]=repeat(select,i,n,det_berth,det_berth_row,pii);
                         if answer == 1
                              det_day_random(i,n)=day_random(i,n);
                              break
                         end
                    end
               end
               %%
               select_Uti(i,n)=ZH_Ori(ZH_day(i,n))/BW(select(i,n))*100;
          end
     end
```

```
        SELECT=[SELECT,select]
        %%
        select_2=select;
        select_2(find(select_2==0))=[];
        select_same=tabulate(select_2);
        select_same_num=find(select_same(:,2) > 1);
            [select_same_row,~]=size(select_same_num);
            time_num=[];
            for g=1:select_same_row
                [select_i,select_j]=find(select == select_same_num(g));
                [select_i_row,~]=size(select_i);
                E=select;
                E(find(E ~= 0))=1 ;
                for h=1:select_i_row
                    time(h)=day_random(select_i(h),select_j(h));
                    time_num=[time_num;time(h),select_i(h),select_j(h)];
                end
                first_time=min(time);
                first=find(time_num(:,1) == first_time);
                for y=1:select_i_row
                    if y ~= first
                        E(select_i(y),select_j(y))=0;
                    end
                end
            end
              %%
        Berth_vacancy=1/((m_row-select_same_row)/216)
        N_Berth_vacancy=[N_Berth_vacancy,Berth_vacancy];
            %%
            E_sum=sum(sum(E));
            E1=E;
            EEE=[EEE,E1];
            [EEE_row,EEE_col]=size(EEE);
            N_EEE_col=[N_EEE_col,EEE_col];
        %%
        select_Uti_sum=sum(sum(select_Uti));
        %%
        select_Uti_cycle=[select_Uti_cycle;select_Uti_sum]
end
%%
select_Uti_cycle1=sum(select_Uti_cycle);
%%
Rec_sum_Berth_vacancy=sum(N_Berth_vacancy);
%%
Planned_completion=sum(sum(scole_end));
%% ttrace
ttrace=select_Uti_cycle1*Rec_sum_Berth_vacancy*sum(scole_end);
trace=sum(ttrace);
N_trace=[N_trace,trace];
aaa=sort(N_trace);
```

Question3
solve3.m

```
clear all
close all
clc
trace=[];
for xm=1:100
%%
load ZH
load ZH_Ori
load BW
load Date
load ZH_wait
%%
pi=[];
for l=1:10
    pi=[pi,0.1];
end
[BW_row,~]=size(BW);
%%
m=[];
day_arrive=[];
N_sum_not0=[];
N_m_row=[];
EEE=[];
E_sum_cycle=[];
select_Uti_cycle=[];
N_EEE_row=[];
N_EEE_col=[];
SELECT=[];
N_Berth_vacancy=[];
min_BC=100;
max_BC=600000;
Docking_time_sum=[];
for xy=1:14
    %%
    select=[];
    day=xy;
    E1=[];
    EEE_row=[];
    EEE_col=[];
    m=[];
    ZH_day=[];
    day_arrive=[];
    sum_not0=[];
    for ijj=1:14
        dayy=rem(day,ijj);
        if dayy == 0
```

```
                day_arrive=[day_arrive,ijj];
            end
        end
        [~,day_arrive_row]=size(day_arrive);
        for q=1:day_arrive_row
            m=[m;find(Date==day_arrive(q))];
        end
        m=sort(m);
        [m_row,~]=size(m);
        N_m_row=[N_m_row,m_row];
        for ii=1:30
            num=1;
            for jj=1:m_row
                cycle=0;
                for ij=1:40
                    if m(jj) == (ii+cycle)
                        ZH_day(ii,num)=m(jj);
                        num=num+1;
                    end
                    cycle=cycle+30;
                end
            end
        end
        day_random=[];
        for ii=1:30
            sum_not0(ii)=sum(sum(ZH_day(ii,:)~=0));
            %%
            day_random(ii,:)=randsrc(1,max(30),[8:17;pi]);
        end
        N_sum_not0=[N_sum_not0;sum_not0];
        %%
        det_berth=[];
        for i=1:30
            if sum_not0(i) == 0;
                continue
            end
            for n=1:sum_not0(i)
                det_berth=[];
                for j=1:BW_row
                    if ZH_Ori(ZH_day(i,n)) < (1.5*BW(j))
                        det_berth=[det_berth,j];
                    end
                end
                [~,det_berth_row]=size(det_berth);
                pii=[];
                for jj=1:det_berth_row
                    pii=[pii,1/det_berth_row];
                end
                BW_random=randsrc(1,1,[1:det_berth_row;pii]);
                if n == 1
```

```
                select(i,1)=det_berth(BW_random);
            end
            det_day_random(i,n)=day_random(i,n);

[scole_CA,scole_CL]=Standard_value(det_day_random,n,i,ZH_day,ZH_wait);
            scole_end(i,n)=scole_CA(i,n)+scole_CL(i,n);
            if sum(scole_end(i,:)) == 0
                for ff=1:ceil(n/2)
                    scole_end(i,ff)=1
                end
            end
            if n > 1
                for cycle1=1:n

[answer,select]=repeat(select,i,n,det_berth,det_berth_row,pii);
                    if answer == 1
                        det_day_random(i,n)=day_random(i,n);
                        break
                    end
                end
            end
            %%
            select_Uti(i,n)=ZH_Ori(ZH_day(i,n))/BW(select(i,n))*100;
               %%
            alpha=(BW(select(i,n))-min_BC)/(max_BC-min_BC)*26000;
            if alpha > 0.5
            Docking_time(i,n)=ZH_Ori(ZH_day(i,n))/alpha;
            else
                Docking_time(i,n)=0.5;
            end
        end
    end
    Docking_time_sum=[Docking_time_sum,Docking_time];
    SELECT=[SELECT,select]
    %%
    select_2=select;
    select_2(find(select_2==0))=[];
    select_same=tabulate(select_2);
    select_same_num=find(select_same(:,2) > 1);
        [select_same_row,~]=size(select_same_num);
        time_num=[];
        for g=1:select_same_row
            [select_i,select_j]=find(select == select_same_num(g));
            [select_i_row,~]=size(select_i);
            E=select;
            E(find(E ~= 0))=1 ;
            for h=1:select_i_row
                time(h)=day_random(select_i(h),select_j(h));
                time_num=[time_num;time(h),select_i(h),select_j(h)];
            end
```

```
                first_time=min(time);
                first=find(time_num(:,1) == first_time);
                for y=1:select_i_row
                    if y ~= first
                        E(select_i(y),select_j(y))=0;
                    end
                end
            end
              %%
        Berth_vacancy=1/((m_row-select_same_row)/216)
        N_Berth_vacancy=[N_Berth_vacancy,Berth_vacancy];
            %%
            E_sum=sum(sum(E));
            E1=E;
            EEE=[EEE,E1];
            [EEE_row,EEE_col]=size(EEE);
            N_EEE_col=[N_EEE_col,EEE_col];
        %%
        select_Uti_sum=sum(sum(select_Uti));
end
%%
Rec_sum_Berth_vacancy=1/sum(N_Berth_vacancy);
%%
N_Docking_time_sum=sum(sum(Docking_time_sum));
%% ttrace 3
ttrace=N_Docking_time_sum*Rec_sum_Berth_vacancy;
trace=[trace,ttrace];
vpa(prod(sym(trace)),7)
end
N_trace=sort(trace,'descend')
plot(N_trace)
```

Question4

solve4.m

```
clear all
close all
clc
w_location=[1:5];
load ZH
load ZH_Ori
load BW
load Date
load ZH_wait
Length=[];
N_Length=[];
N_evaluating=[];
N_eva=[];
N_end=[];
N_SELECT=[];
NLength=0;
```

```
BW_length=[100,200,300,2000,3000,7000,10000,20000,24000,25000,...
    30000,40000,50000,60000,65000,80000,90000,100000,130000,...
    150000,170000,200000,600000]
[~,BW_length_col]=size(BW_length);
for i=1:BW_length_col
    e=find(BW == BW_length(i));
    [e_row,e_col]=size(e);
    Length=[Length,length(e)];
    NLength=NLength+length(e);
    N_Length=[N_Length,NLength];
    for k=1:e_row
        if NLength <= 44
            BW(e(k),2)=1;
        elseif 44 < NLength && NLength <= 82
            BW(e(k),2)=2;
        end
        if 82 < NLength && NLength <= 131
            BW(e(k),2)=3;
        elseif 131 < NLength && NLength <= 171
            BW(e(k),2)=4;
        end
        if 171 < NLength && NLength <= 216
            BW(e(k),2)=5;
        end
    end
end
for gggg=1:100
%%
pi=[];
for l=1:10
    pi=[pi,0.1];
end
[BW_row,~]=size(BW);
%%
m=[];
day_arrive=[];
N_sum_not0=[];
N_m_row=[];
EEE=[];
E_sum_cycle=[];
select_Uti_cycle=[];
N_EEE_row=[];
N_EEE_col=[];
SELECT=[];
N_Berth_vacancy=[];
BC_min=100;
BC_max=600000;
CC_min=105;
CC_max=156621;
beta0=1;
```

```
for xhy=1:14
    %%
    select=[];
    day=xhy;
    E1=[];
    EEE_row=[];
    EEE_col=[];
    m=[];
    ZH_day=[];
    day_arrive=[];
    sum_not0=[];
    for ijj=1:14
        dayy=rem(day,ijj);
        if dayy == 0
            day_arrive=[day_arrive,ijj];
        end
    end
    [~,day_arrive_row]=size(day_arrive);
    for q=1:day_arrive_row
        m=[m;find(Date==day_arrive(q))];
    end
    m=sort(m);
    [m_row,~]=size(m);
    N_m_row=[N_m_row,m_row];
    for ii=1:30
        num=1;
        for jj=1:m_row
            cycle=0;
            for ij=1:40
                if m(jj) == (ii+cycle)
                    ZH_day(ii,num)=m(jj);
                    num=num+1;
                end
                cycle=cycle+30;
            end
        end
    end
    day_random=[];
    for ii=1:30
        sum_not0(ii)=sum(sum(ZH_day(ii,:)~=0));
        %%
        day_random(ii,:)=randsrc(1,max(30),[8:17;pi]);
    end
    N_sum_not0=[N_sum_not0;sum_not0];
    %%
    det_berth=[];
    for i=1:30
        if sum_not0(i) == 0;
            continue
        end
```

```
            for n=1:sum_not0(i)
                det_berth=[];
                for j=1:BW_row
                    if ZH_Ori(ZH_day(i,n)) < (1.5*BW(j))
                        det_berth=[det_berth,j];
                    end
                end
                [~,det_berth_row]=size(det_berth);
                pii=[];
                for jj=1:det_berth_row
                    pii=[pii,1/det_berth_row];
                end
                BW_random=randsrc(1,1,[1:det_berth_row;pii]);
                if n == 1
                    select(i,1)=det_berth(BW_random);
                end
                det_day_random(i,n)=day_random(i,n);

[scole_CA,scole_CL]=Standard_value(det_day_random,n,i,ZH_day,ZH_wait);
                scole_end(i,n)=scole_CA(i,n)+scole_CL(i,n);
                if sum(scole_end(i,:)) == 0
                    for ff=1:ceil(n/2)
                        scole_end(i,ff)=1
                    end
                end
                if n > 1
                    for cycle1=1:n

[answer,select]=repeat(select,i,n,det_berth,det_berth_row,pii);
                        if answer == 1
                            det_day_random(i,n)=day_random(i,n);
                            break
                        end
                    end
                end
                %%
                w_location(i,n)=randperm(5,1);
                beta=beta0*(ZH_Ori(ZH_day(i,n))-CC_min)/(CC_max-CC_min);
                alpha(i,n)=(BW(select(i,n))-BC_min)/(BC_max-
BC_min)*ZH_Ori(ZH_day(i,n))*beta;
                w_BC(i,n)=BW(select(i,n),2);
                evaluating_indicator(i,n)=w_location(i,n)+alpha(i,n)+w_BC(i,n);
                %%
                select_Uti(i,n)=ZH_Ori(ZH_day(i,n))/BW(select(i,n))*100;
            end
        end
        N_evaluating=[N_evaluating,evaluating_indicator];
        SELECT=[SELECT,select]
        %%
        select_2=select;
```

```
        select_2(find(select_2==0))=[];
        select_same=tabulate(select_2);
        select_same_num=find(select_same(:,2) > 1);
            [select_same_row,~]=size(select_same_num);
            time_num=[];
            for g=1:select_same_row
                [select_i,select_j]=find(select == select_same_num(g));
                [select_i_row,~]=size(select_i);
                E=select;
                E(find(E ~= 0))=1 ;
                for h=1:select_i_row
                    time(h)=day_random(select_i(h),select_j(h));
                    time_num=[time_num;time(h),select_i(h),select_j(h)];
                end
                first_time=min(time);
                first=find(time_num(:,1) == first_time);
                for y=1:select_i_row
                    if y ~= first
                        E(select_i(y),select_j(y))=0;
                    end
                end
            end
              %%
        Berth_vacancy=1/((m_row-select_same_row)/216)
        N_Berth_vacancy=[N_Berth_vacancy,Berth_vacancy];
            %%
            E_sum=sum(sum(E));
            E1=E;
            EEE=[EEE,E1];
            [EEE_row,EEE_col]=size(EEE);
            N_EEE_col=[N_EEE_col,EEE_col];
        %%
        select_Uti_sum=sum(sum(select_Uti));
        %%
         select_Uti_cycle=[select_Uti_cycle;select_Uti_sum]
    end
    %%
    select_Uti_cycle1=sum(select_Uti_cycle);
    %%
    N_evaluating_indicator=sum(sum(N_evaluating));
    %%
    END=N_evaluating_indicator*select_Uti_cycle1;
    N_SELECT=[N_SELECT,select_Uti_cycle1];
    N_eva=[N_eva,N_evaluating_indicator];
    N_end=[N_end,END];
    end
    plot(N_end)
    end
```

Question5

solve5.m

```
clear all
close all
clc
w_location=[1:5];
load ZH
load ZH_Ori
load BW
load Date
load ZH_wait
Length=[];
N_Length=[];
N_evaluating=[];
N_eva=[];
N_end=[];
N_SELECT=[];
NLength=0;
BW_length=[100,200,300,2000,3000,7000,10000,20000,24000,25000,...
    30000,40000,50000,60000,65000,80000,90000,100000,130000,...
    150000,170000,200000,600000]
[~,BW_length_col]=size(BW_length);
for i=1:BW_length_col
    e=find(BW == BW_length(i));
    [e_row,e_col]=size(e);
    Length=[Length,length(e)];
    NLength=NLength+length(e);
    N_Length=[N_Length,NLength];
    for k=1:e_row
        if NLength <= 44
            BW(e(k),2)=1;
        elseif 44 < NLength && NLength <= 82
            BW(e(k),2)=2;
        end
        if 82 < NLength && NLength <= 131
            BW(e(k),2)=3;
        elseif 131 < NLength && NLength <= 171
            BW(e(k),2)=4;
        end
        if 171 < NLength && NLength <= 216
            BW(e(k),2)=5;
        end
    end
end
%%
pi=[];
for l=1:10
    pi=[pi,0.1];
end
[BW_row,~]=size(BW);
```

```
%%
m=[];
day_arrive=[];
N_sum_not0=[];
N_m_row=[];
EEE=[];
E_sum_cycle=[];
select_Uti_cycle=[];
N_EEE_row=[];
N_EEE_col=[];
SELECT=[];
N_Berth_vacancy=[];
BC_min=100;
BC_max=600000;
CC_min=105;
CC_max=156621;
beta0=1;
for run=1:100
    for xhy=1:14
        %%
        select=[];
        day=xhy;
        E1=[];
        EEE_row=[];
        EEE_col=[];
        m=[];
        ZH_day=[];
        day_arrive=[];
        sum_not0=[];
        for ijj=1:14
            dayy=rem(day,ijj);
            if dayy == 0
                day_arrive=[day_arrive,ijj];
            end
        end
        [~,day_arrive_row]=size(day_arrive);
        for q=1:day_arrive_row
            m=[m;find(Date==day_arrive(q))];
        end
        m=sort(m);
        [m_row,~]=size(m);
        N_m_row=[N_m_row,m_row];
        for ii=1:30
            num=1;
            for jj=1:m_row
                cycle=0;
                for ij=1:40
                    if m(jj) == (ii+cycle)
                        ZH_day(ii,num)=m(jj);
                        num=num+1;
```

```
                        end
                        cycle=cycle+30;
                    end
                end
            end
            day_random=[];
            for ii=1:30
                sum_not0(ii)=sum(sum(ZH_day(ii,:)~=0));
                %%
                day_random(ii,:)=randsrc(1,max(30),[8:17;pi]);
            end
            N_sum_not0=[N_sum_not0;sum_not0];
            %%
            det_berth=[];
            for i=1:30
                if sum_not0(i) == 0;
                    continue
                end
                for n=1:sum_not0(i)
                    det_berth=[];
                    for j=1:BW_row
                        if ZH_Ori(ZH_day(i,n)) < (1.5*BW(j))
                            det_berth=[det_berth,j];
                        end
                    end
                    [~,det_berth_row]=size(det_berth);
                    pii=[];
                    for jj=1:det_berth_row
                        pii=[pii,1/det_berth_row];
                    end
                    BW_random=randsrc(1,1,[1:det_berth_row;pii]);
                    if n == 1
                        select(i,1)=det_berth(BW_random);
                    end
                    det_day_random(i,n)=day_random(i,n);

[scole_CA,scole_CL]=Standard_value(det_day_random,n,i,ZH_day,ZH_wait);
                    scole_end(i,n)=scole_CA(i,n)+scole_CL(i,n);
                    if sum(scole_end(i,:)) == 0
                        for ff=1:ceil(n/2)
                            scole_end(i,ff)=1
                        end
                    end
                    if n > 1
                        for cycle1=1:n

[answer,select]=repeat(select,i,n,det_berth,det_berth_row,pii);
                            if answer == 1
                                det_day_random(i,n)=day_random(i,n);
                                break
```

```
                    end
                end
            end
            %%
            if xhy <= 5
                num_berth=20;
                num_berth_select=num_berth;
            elseif 5 < xhy && xhy <= 7
                num_berth=20;
                numu_berth_relax=randperm(20,1);
                num_berth_select=num_berth-numu_berth_relax;
            end
            if 8 < xhy && xhy <= 12
                num_berth=20;
                num_berth_select=num_berth;
            elseif 12 < xhy && xhy <= 14
                num_berth=20;
                numu_berth_relax=randperm(20,1);
                num_berth_select=num_berth-numu_berth_relax;
            end
            if num_berth_select == 0;
                num_berth_select=1;
            end
            port_select(i,n)=randperm(num_berth_select,1);
            %%
            w_location(i,n)=randperm(5,1);
            beta=beta0*(ZH_Ori(ZH_day(i,n))-CC_min)/(CC_max-CC_min);
            alpha(i,n)=(BW(select(i,n))-BC_min)/(BC_max-
BC_min)*ZH_Ori(ZH_day(i,n))*beta;
            w_BC(i,n)=BW(select(i,n),2);
            evaluating_indicator(i,n)=w_location(i,n)+alpha(i,n)+w_BC(i,n);
            %%
            select_Uti(i,n)=ZH_Ori(ZH_day(i,n))/BW(select(i,n))*100;
        end
    end
    N_evaluating=[N_evaluating,evaluating_indicator];
    SELECT=[SELECT,select]
    %%
    select_2=select;
    select_2(find(select_2==0))=[];
    select_same=tabulate(select_2);
    select_same_num=find(select_same(:,2) > 1);
    [select_same_row,~]=size(select_same_num);
    time_num=[];
    for g=1:select_same_row
        [select_i,select_j]=find(select == select_same_num(g));
        [select_i_row,~]=size(select_i);
        E=select;
        E(find(E ~= 0))=1 ;
        for h=1:select_i_row
```

```
                    time(h)=day_random(select_i(h),select_j(h));
                    time_num=[time_num;time(h),select_i(h),select_j(h)];
                end
                first_time=min(time);
                first=find(time_num(:,1) == first_time);
                for y=1:select_i_row
                    if y ~= first
                        E(select_i(y),select_j(y))=0;
                    end
                end
            end
            %%
            Berth_vacancy=1/((m_row-select_same_row)/216)
            N_Berth_vacancy=[N_Berth_vacancy,Berth_vacancy];
            %%
            E_sum=sum(sum(E));
            E1=E;
            EEE=[EEE,E1];
            [EEE_row,EEE_col]=size(EEE);
            N_EEE_col=[N_EEE_col,EEE_col];
            %%
            select_Uti_sum=sum(sum(select_Uti));
            %%
            select_Uti_cycle=[select_Uti_cycle;select_Uti_sum]
            %% This part is used to save the three data
            %           save(['***','Day： ',num2str(xhy)],'ZH_day');
            %           save(['***','Day： ',num2str(xhy)],'select');
            %       save(['***','Day： ',num2str(xhy)],'port_select');
        end
        %%
        select_Uti_cycle1=sum(select_Uti_cycle);
        %%
        N_evaluating_indicator=sum(sum(N_evaluating));
        %%
        END=N_evaluating_indicator*select_Uti_cycle1;
        N_SELECT=[N_SELECT,select_Uti_cycle1];
        N_eva=[N_eva,N_evaluating_indicator];
        N_end=[N_end,END];
end
```

Standard_value.m

```
function
[scole_CA,scole_CL]=Standard_value(det_day_random,n,i,ZH_day,ZH_wait)
standard_get=12;
standard_leave=18;
if (det_day_random(i,n)-standard_get) == 0
        scole_CA(i,n)=1;
elseif (det_day_random(i,n)-standard_get) < 0
        scole_CA(i,n)=0;
```

```
else
    scole_CA(i,n)=-1;
end
time_leave(i,n)=det_day_random(i,n)+ZH_wait(ZH_day(i,n));
if standard_leave-time_leave(i,n) < 0
    scole_CL(i,n)=1;
else
    scole_CL(i,n)=0;
end
end
```

repeat.m

```
function [answer,select]=repeat(select,i,n,det_berth,det_berth_row,pii)
for hyh=1:(n-1)
    BW_random1=randsrc(1,1,[1:det_berth_row;pii]);
    if select(i,hyh) ~= BW_random1
        select(i,n)=det_berth(BW_random1);
        answer=1;
    else
        answer=0;
    end
end
```